Universidad Carlos III de Madrid

Escuela Politécnica Superior

Ingeniería de Telecomunicación

PROYECTO FIN DE CARRERA

AUTOORGANIZACIÓN DE COLECCIONES DE DOCUMENTOS MEDIANTE TÉCNICAS DE AGRUPAMIENTO ESPECTRAL

Autor: Elena Rábanos Iruela
Tutor: Dr. Emilio Parrado Hernández

Febrero de 2011

Título: *AUTOORGANIZACIÓN DE COLECCIONES DE DOCUMENTOS MEDIANTE TÉCNICAS DE AGRUPAMIENTO ESPECTRAL.*

Autor: *ELENA RÁBANOS IRUELA*

Tutor: *Dr. EMILIO PARRADO HERNÁNDEZ*

La defensa del presente Proyecto Fin de Carrera se realizó el día 10 de Febrero de 2011; siendo calificada por el siguiente tribunal:

Presidenta: *Vanessa Gómez Verdejo*

Secretaria: *Sara Pino Povedano*

Vocal *Clara Marina Sanz García*

Agradecimientos

En primer lugar quiero dar las gracias al Prof. Dr. Emilio Parrado Hernández por su gran labor como tutor en el desarrollo de el proyecto fin de carrera. Agradezco su apoyo, dedicación y paciencia en este largo camino.

No puedo expresar con palabras el agradecimiento que le debo a mi familia, por su apoyo y confianza en este tiempo, especialmente a mis padres, Mariví y Miguel, a mi hermano Alberto, a mis tíos, Rafa y Nuria, y mi prima Marisa. Gracias porque sin ellos no hubiera podido llevar a cabo este trabajo.

Quiero agradecer también a todos los compañeros que he tenido en la carrera, con los que hemos pasado alegrías y penas. Especialmente quiero dar las gracias a Saray, Sara, Lucía y Pilar todo el apoyo mostrado tanto en este tiempo con el proyecto como en los años de carrera.

Para mi el baloncesto ha sido pieza fundamental en mi vida y ha sido una via de escape cuando las cosas en la universidad estaban un poco complicadas. Por ello agradezco a todas las compañeras de equipo y amigas que han estado apoyándome continuamente en la cancha cuando lo he necesitado. En todo este tiempo son muchos los equipos donde he estado, pero agradezco primero a mis compañeras del equipo de la Universidad Carlos III: Luisa, Inés, Esther, Lourdes, Ali y Arantxa. Quiero destacar también a mis Espartanas: Laura, Nat, Sandra, María, Almu y Cris, que tanto cariño y apoyo me bridaron en esa temporada y que aún siguen haciendo. Quiero agradecer el apoyo que me dan a mis compañeras de mi equipo actual, espero que a partir a ahora pueda ir a entrenar algo más.

Por último quiero dar las gracias por el apoyo y el cariño que he recibido de mi gente de la Escuela de Idiomas de Barajas, especialmente a Laura, Isabel, Valentina, Julia, Charo, Inma, Lore, Rocío, Sara, Carmen, Pilar y Gloria.

Puedo aceptar el fracaso, pero no puedo aceptar no tratar de intentarlo.

Quien dice que juega al límite es porque lo tiene.

Michael Jordan

Resumen

El objetivo de este proyecto es el estudio de las prestaciones ofrecidas por los algoritmos de *clustering* espectral en estudio para el problema de auto-organización de documentos. Se trata por tanto de un problema de minería de datos utilizando métodos de *clustering* avanzado para su resolución. Una vez expuesto el marco teórico y tecnológico en el que se fundamenta este proyecto, se realiza un trabajo experimental mediante la implementación de dichos algoritmos sobre la plataforma Matlab, para así poder obtener resultados concretos de las prestaciones de dicho algoritmos sobre las diferentes bases de datos de documentos utilizadas.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

El primer capítulo del proyecto se dedica a presentar el marco tecnológico en el que está encuadrado nuestro problema de autoorganización de documentos. Dicha tarea se trata como un problema de minería de datos mediante técnicas de agrupamiento o *clustering*. A continuación, exponemos los fundamentos de dichas disciplinas, minería de datos y *clustering*, y cómo a partir de ellos nos planteamos unos objetivos para dar solución a nuestro problema de partida. Por último describiremos la estructura del documento.

1.1. Minería de datos

Debido al gran avance de la tecnología, la capacidad para generar y almacenar datos ha crecido enormemente lo que ha llevado a desarrollar técnicas de análisis de todos estos datos para obtener un conocimiento de ellos. Este campo de investigación de conoce como KDD (*Knowledge Discovery in Databases*) [10] dedicado a desarrollar procesos de descubrimento de conocimiento en grandes volúmenes de datos. Actualmente este nombre se ha sustituido por Minería de Datos debido a la generalización de una de las etapas que forma parte de todo proceso KDD, en donde se aplican las técnicas y algoritmos de descubrimiento.

Por tanto definimos minería de datos como aquel proceso de extracción de información desconocida con anterioridad, válida y potencialmente útil de grandes bases de datos para usarla con posterioridad para la toma de decisiones. Visto su gran potencial, la minería de datos es actualmente muy utilizada en el mundo empresarial y de marketing para obtener información a partir de sus bases de datos y poder tomar decisiones de negocio.

Algunos ejemplos de procesos de minería de datos que ponen en práctica las empresas son:

- Segmentación de la cartera de clientes, es decir encontrar grupos de clientes con características similares.
- Detectar los productos que con más frecuencia se compran juntos, con el objetivo de promover ofertas o descuentos o vender productos en pack.
- Encontrar el perfil del comprador del producto X.
- Predecir el nivel de morosidad de un cliente
- Detectar los clientes que están cometiendo acciones fraudulentas.
- Encontrar los síntomas de enfermedades que más a menudo aparecen juntas.

El proceso de KDD consta de varias fases:

1. **Selección de objetivos:** En esta fase se estudia el problema y se fija la meta del proceso. Si se realiza un buen planteamiento del problema, el resto de etapas resultarán más sencillas.
2. **Preparación de los datos:** Inicialmente se identifican las bases de datos y se selecciona el subconjunto de datos necesarios para la aplicación. Posteriormente se realiza el procesado de los datos, fase en la que es necesario estudiar los datos, entendiendo el significado de los atributos y detectando posibles datos erróneos, repetidos o con formatos diferentes.
3. **Transformación de los datos:** Se selecciona el algoritmo de minería de datos que vamos a aplicar en la siguiente etapa y los parámetros de entrada que recibe dicho algoritmo. Cada algoritmo requiere un formato diferente en los datos de entrada, por tanto debemos transformar los datos para que se ajusten al formato de entrada del algoritmo seleccionado.
4. **Aplicación técnicas o algoritmos de minería de datos:** Es la etapa principal del proceso, donde se aplican los algoritmos de análisis de datos según el tipo de problema a resolver.

5. **Análisis de los resultados:** En esta etapa se interpretan y evalúan los resultados obtenidos en la etapa anterior. En un contexto empresarial, por ejemplo, a partir de los resultados obtenidos se tomarán las decisiones de negocio necesarias.

Como hemos comentado anteriormente, existen diferentes problemas de minería de datos y los podemos clasificar en dos grandes grupos:

- **Problemas descriptivos:** El objetivo de este tipo de problemas es encontrar una descripción de los datos de estudio. A su vez estos problemas pueden ser de dos tipos:
  - **Segmentación de los datos:** En este tipo de problemas la meta es encontrar grupos homogéneos en los datos de estudio. A estos problemas también se les denomina problemas de aprendizaje no supervisado y se aplican algoritmos de *clustering* para conseguirlo. Un ejemplo típico de este tipo de problemas es la segmentación de los clientes de una empresa o un servicio.
  - **Búsqueda de asociaciones entre los datos:** El objetivo de este tipo de problemas es encontrar relaciones entre los valores de los atributos de los datos. Se suelen usar técnicas de búsqueda de patrones. Un ejemplo de este tipo de problemas es realizar un análisis de la cesta de la compra, buscando relaciones entre los productos comprados por cada cliente.
- **Problemas predictivos:** El objetivo de este tipo de problemas es obtener un modelo para posteriormente poder predecir comportamientos. Se les denomina problemas de aprendizaje supervisado. Existen diferentes tipos de problemas predictivos:
  - **Clasificación de los datos de entrada**: Se tata de problemas en los que la variable a predecir tiene un número finito de valores (variable categórica). Se suelen utilizar algoritmos basados en árboles de decisión y redes neuronales. Un ejemplo de este tipo de problemas sería encontrar un modelo que a la vista del histórico de clientes clasificados como *buenos*, *regulares* y *malos*, determine de qué tipo es un cliente nuevo.
  - **Predicción de valores**: En este caso la variable a predecir es numérica. Se suele usar técnicas de regresión, tanto lineal como no lineal. En este tipo de problemas podríamos encontrar un modelo que pudiera predecir la probabilidad de que un cliente devuelva el crédito que ha solicitado.

1.2. Clustering

Clustering [3] es el proceso de agrupar los datos en clases o grupos, de modo que los objetos dentro de un *cluster* (agrupación) tienen una alta similitud entre ellos, pero son muy diferentes a los objetos de otros grupos. La comparación entre objetos se hace en base a los atributos que los describen.

Este tipo de análisis es utilizado en numerosas aplicaciones, tales como la investigación de mercado, el reconocimiento de patrones, el análisis de datos y el procesamiento de imágenes. Por ejemplo, en el caso de aplicaciones de marketing, mediante procesos de *clustering* podemos descubrir grupos distintos de clientes en las bases de datos y caracterizarlos en función de sus patrones de compra.

En todo algoritmo de *clustering* hay que tener en cuenta dos asuntos importantes:

- Medida de similitud o disimilitud, según el algoritmo, para poder realizar la comparación entre objetos.
- Optimización de un objetivo, de tal manera que encontremos el *clustering* en función de una maximización de la similitud entre los objetos de un *cluster* y minimización de la similitud entre los objetos de diferentes *clusters*.

Como hemos comentado la comparación entre los datos de entrada se realiza en función de una medida de similitud o disimilitud, según la implementación de cada algoritmo. Hablamos de medida de similitud cuando al comparar dos datos, éstos serán más parecidos cuanto mayor sea el valor de dicha medida entre ellos. Un ejemplo de este tipo de medida es la operación coseno entre dos datos, $cos(x, y)$, en donde la similitud es máxima cuando el valor del coseno sera igual a 1 y por el contrario, los datos serán más diferentes cuanto menor sea este valor. Por otro lado, hablamos de medida de disimilitud cuando al comparar dos datos, los objetos son más parecidos cuanto menor es esa medida entre ellos. La medida de distancias entre objetos es un tipo de medida de disimilitud, puesto que cuanto más alejados estén los objetos, menos parecidos son dichos objetos. La medida de distancia más conocida es la distancia Euclídea:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

siendo $x = (x_1, x_2, \ldots, x_n)$ e $y = (y_1, y_2, \ldots, y_n)$ los objetos para los que se calcula la distancia euclídea.

Existen diversos algoritmos de clustering [3] como pueden ser el K-medias, el LBG (Linde-Buzo-Gray) y el ISODATA (*Iterative Self-Organizing Data Analysis Technique Algorithm*). Todos ellos podemos denominarles algoritmo de *clustering* básico, donde unos algoritmos se difieren de los otros en base a su medida de similitud o disimilitud entre los datos y los criterios de optimización en la búsqueda del mejor *clustering*. Por otro lado, existen otro tipo de algoritmos de *clustering* denominados algoritmos de *clustering* espectral, que dan un paso más con respecto a los algoritmos anteriormente mencionados. Este tipo de algoritmos calculan el agrupamiento de los datos en base a la información espectral de los datos, es decir mediante la información aportada por los autovalores y autovectores de las medidas de similitudes calculadas entre los datos.

El estudio de este tipo de métodos de *clustering* espectral se desarrollará en el próximo capítulo, pues es parte del objeto de estudio de este proyecto.

1.3. Autoorganización de documentos

Mediante la autoorganización automática de documentos podemos agrupar grandes volúmenes de documentos en grupos categorizados, en los que aquellos documentos pertenecientes al mismo grupo tendrán temáticas similares. Disponer de documentos agrupados nos permite tener una base de datos de documentos categorizadas por temáticas con el fin de poder tener acceso a ellos de manera inmediata y además poder encontrar documentos relacionados con uno de partida de una manera más rápida y eficiente que si tuviéramos que buscar en toda la base de datos.

El poder realizar esta tarea de manera automática genera un gran ahorro de tiempo, puesto que de otra manera habría que leer los documentos y decidir cuáles son parecidos entre sí. Aquí puede influir también la subjetividad de la persona que lea y etiquete. Realizar la organización de grandes volúmenes de documentos de manera manual resulta prácticamente inviable, porque además puede ser una tarea que no tenga fin, dada la cantidad de información que se genera en la actualidad. Por tanto se requiere de un sistema automático de autoorganización de documentos con el fin de dar solución a este problema.

1.4. Objetivos

El objetivo principal de este proyecto es el estudio de las prestaciones ofrecidas por los algoritmos de *clustering* espectral en la resolución del problema de autoorganización de documentos. Para ello es necesario contextualizar el problema en un entorno de minería de datos, estudiar y analizar los diferentes algoritmos de *clustering* espetral que van a ser ulilizados y determinar unas medidas para la evaluación de la calidad de nuestro organizador de documentos. Posteriormente, se realizarán diferentes simulaciones sobre unas bases de datos de documentos para así obtener unos resultados que nos indiquen qué algoritmo de *clustering* espectral es más adecuado para este problema.

1.5. Estructura del documento

Este documento está en divido en 5 capítulos, cada uno de ellos focalizado en un aspecto diferente del desarrollo del proyecto.

Como hemos visto, en el capítulo 1 se realiza un presentación de marco tecnológico en el que está encuadrado el problema a resolver por este proyecto.

En el capítulo 2, se describen los algoritmos de *clustering* espectral utilizados para evaluar las prestaciones de nuestro autoorganizador de documentos. Por tanto, este capítulo constituye el estado del arte en métodos de *clustering* espectral.

Una vez presentados los algoritmos que vamos a utilizar, en el capítulo 3 vamos a estudiar de qué manera vamos a parametrizar los documentos para poder aplicarles dichos algoritmos y cumplir nuestro objetivo de implementar un autoorganizador de documentos. En este capítulo también describimos qué medida de calidad vamos a utilizar para evaluar las prestaciones de nuestro autoorganizador de documentos. Esta medida está basada en la información mutua.

En el capítulo 4 se describen qué bases de datos de documentos hemos utilizado en nuestro trabajo experimental y mostraremos los resultados conseguidos para cada una de las bases de datos, evaluándolos y llegando a la conclusión de qué algoritmo consigue mejores prestaciones para la tarea de autoorganizador de documentos.

Finalmente, el capítulo 5 recoge las conclusiones extraídas del trabajo realizado en este proyecto así como la presentación de posibles líneas futuras como mejora del las investigaciones realizadas.

Capítulo 2

Revisión del Estado del Arte

Como hemos comentado en el capítulo anterior, nuestro problema de autoorganización de documentos se basa en realizar un ejercicio de *clustering* sobre los datos de entrada, de tal manera que agrupemos los datos en grupos homogéneos, es decir, con características parecidas.

El objetivo de este proyecto es evaluar algoritmos de *clustering* avanzados, basados en soluciones espectrales, por tanto, a continuación se muestran los cuatro algoritmos que son objeto de estudio. Previamente se detalla la notación común seguida por dichos algoritmos, explicando qué representa cada variable utilizada en el desarrollo de los mismos.

2.1. Elementos de un problema de agrupamiento

A continuación se va detallar la notación utilizada en cada uno de los métodos de *clustering* espectral:

- Número de datos de entrada: N
- Datos de entrada: matriz X de dimensiones $N \times l$, por tanto son N objetos de entrada cada uno de ellos con l atributos.
- Número de *clusters*: R
- Matriz de clustering: C. Esta matriz tiene dimensiones $N \times R$, registra para cada dato de entrada el *cluster* al que pertenece, seleccionando la columna que le corresponde a dicho *cluster*.

- Matriz de similitud: W. Cada elemento de la matriz, W_{ij}, representa la similitud entre el objeto i y el objeto j.
- Centroide: μ. Representa las coordenadas de un centroide. Puesto que hay R *clusters* tendremos R centroides.
- Función de coste del *clustering*: $J(C)$
- Autovectores: Y
- Autovalores: λ
- Particiones: cuando son dos particiones $\Rightarrow$ A y B, cuando son R particiones $\Rightarrow$ matriz E. Almacenan la información de qué datos pertenecen a cada partición.

2.2. *Clustering* en un espacio de características

Este método hace uso de los métodos de *kernel* [7] como mecanismo para proyectar los datos de entrada a otro espacio denominado espacio de características donde se aplican algoritmos de agrupamiento.

El uso de métodos de *kernel* para realizar agrupamientos es muy natural, ya que mediante la función de *kernel* definimos similitudes entre pares de datos, lo que proporciona toda la información necesaria para realizar el agrupamiento y evaluar la calidad del mismo. A continuación se van a presentar una serie de algoritmos que usan funciones *kernel* como medida de similitud.

2.2.1. Medida de Calidad del agrupamiento

Dado un conjunto de datos sin etiquetar: $X = \{x_1, ..., x_N\}$ y un número de grupos R, queremos encontrar una función $f : X \to \{1, 2, ..., R\}$ que asigne cada punto a uno de los R grupos.

Esta partición de los datos debe elegirse evaluando un criterio de calidad sobre todos los posibles agrupamientos. El agrupamiento será elegido teniendo en cuenta la optimización de la siguiente función:

$$f^* = \arg\min_f \sum_{i,j:f_i=f(x_i)=f(x_j)=f_j} \|\phi(x_i) - \phi(x_j)\|^2 \tag{2.1}$$

que encuentra aquella asociación f que minimiza el cuadrado de la distancia entre los puntos pertenecientes al mismo *cluster* y donde asumimos que ϕ es la función proyección al espacio de características F, en donde la función *kernel* calcula el producto escalar entre las funciones ϕ de dos puntos:

$$\kappa\left(x_i, x_j\right) = \left\langle \phi\left(x_i\right), \phi\left(x_j\right)\right\rangle \tag{2.2}$$

Este primer criterio no es completo, pues no tiene en cuenta la distancia entre los demás clusters, por tanto es necesario modificar el criterio anterior:

$$\min_f \left\{ \sum_{i,j:f_i=f_j} \left\|\phi\left(x_i\right) - \phi\left(x_j\right)\right\|^2 - \beta \sum_{i,j:f_i\neq f_j} \left\|\phi\left(x_i\right) - \phi\left(x_j\right)\right\|^2 \right\} \tag{2.3}$$

donde observamos que cuanto menor sea la distancia entre los puntos del mismo *cluster* y mayor la distancia entre puntos de *clusters* diferentes, la calidad del agrupamiento será mucho mejor.

Si reescribimos el segundo término de la expresión anterior:

$$\begin{aligned} \sum_{i,j:f_i\neq f_j} \left\|\phi\left(x_i\right) - \phi\left(x_j\right)\right\|^2 &= \sum_{i,j=1}^{N} \left\|\phi\left(x_i\right) - \phi\left(x_j\right)\right\|^2 - \sum_{i,j:f_i=f_j} \left\|\phi\left(x_i\right) - \phi\left(x_j\right)\right\|^2 \\ &= \Psi - \sum_{i,j:f_i=f_j} \left\|\phi\left(x_i\right) - \phi\left(x_j\right)\right\|^2 \end{aligned}$$

donde Ψ es una constante proporcionada por el conjunto de datos. Podemos reescribir el criterio anterior de la siguiente manera:

$$\begin{aligned} & \min_f \left\{ \sum_{i,j:f_i=f_j} \left\|\phi\left(x_i\right) - \phi\left(x_j\right)\right\|^2 - \beta \sum_{i,j:f_i\neq f_j} \left\|\phi\left(x_i\right) - \phi\left(x_j\right)\right\|^2 \right\} \\ = & \min_f \left\{ (1+\beta) \sum_{i,j:f_i=f_j} \left\|\phi\left(x_i\right) - \phi\left(x_j\right)\right\|^2 - \beta\Psi \right\} \end{aligned}$$

Este criterio resuelve las dos optimizaciones necesarias: minimizando la distancia entre puntos internos del *cluster*, maximizamos también la distancia entre *clusters*.

Si seguimos desarrollando el primer criterio de optimización:

$$
\begin{aligned}
opt &= \sum_{i,j:f_i=f_j} \|\phi(x_i) - \phi(x_j)\|^2 \\
&= \sum_{k=1}^{R} \sum_{i:f_i=k} \sum_{j:f_j=k} \langle \phi(x_i) - \phi(x_j), \phi(x_i) - \phi(x_j) \rangle \\
&= \sum_{k=1}^{R} 2 \left(\left| f^{-1}(k) \right| \sum_{i:f_i=k} \kappa(x_i, x_i) - \sum_{i:fi=k} \sum_{j:f_j=k} \kappa(x_i, x_j) \right) \\
&= \sum_{k=1}^{R} 2 \left| f^{-1}(k) \right| \sum_{i:f_i=k} \|\phi(x_i) - \mu_k\|^2
\end{aligned}
$$

donde:

- $|f^{-1}(k)|$ denota el numero de puntos pertenecientes al cluster k
- μ_k es el centro de masas o centroide del *cluster* k y tiene la siguiente expresión:

$$\mu_k = \frac{1}{|f^{-1}(k)|} \sum_{i \in f^{-1}(k)} \phi(x_i) \tag{2.4}$$

Por tanto, utilizando la expresión anterior en el criterio de optimización:

$$
\begin{aligned}
f &= \arg\min_f \sum_{i,j:f_i=f(x_i)=f(x_j)=f_j} \|\phi(x_i) - \phi(x_j)\|^2 \\
&= \arg\min_f \sum_{k=1}^{R} \left(\sum_{i:f_i=k} \|\phi(x_i) - \mu_k\|^2 \right) \\
&= \arg\min_f \sum_{i=1}^{N} \left\| \phi(x_i) - \mu_{f(x_i)} \right\|^2
\end{aligned}
$$

Estrategia de optimización del agrupamiento

La estrategia de optimización viene dada por el siguiente proceso:

- Entradas: $X = \{x_1, ..., x_N\}$ y número de *clusters* R. Aplicamos función *kernel*.

- Proceso: Calculamos los centroides de los R grupos $\{\mu_1, \ldots, \mu_R\}$ $\mu = \arg\min_\mu \sum_{i=1}^{N} \min_{1 \leq k \leq R} \|\phi(x_i) - \mu_k\|^2$
- Salidas: Determinar en que *cluster* está cada punto $f(\cdot) = \arg\min_{1 \leq k \leq R} \|\phi(\cdot) - \mu_k\|$

Desafortunadamente este problema no es convexo. La tarea de comprobación para saber si existe una solución mejor que algún umbral se convierte en un problema NP-completo. Esta clase de problemas se dice que no puede ser resuelto en un tiempo polinómico y estaríamos forzados a utilizar algoritmos aproximados para encontrar una solución cercana a la óptima.

A continuación se van a describir dos aproximaciones para solucionar el problema anterior. La primera utilizará un método iterativo para encontrar un mínimo local de la función de coste. La segunda utilizará una relajación de la función de coste para proporcionar una aproximación convexa que será globalmente optimizada.

El método iterativo nos llevará al conocido método de K-medias mientras que el método de relajación nos llevará a algoritmos de *clustering* espectral.

2.2.2. Solución iterativa: Kernel K-medias

El objetivo de este algoritmo es identificar el centro de masas o centroide de cada *cluster*. Este conjunto de centroides $\{\mu_1, \mu_2, ..., \mu_R\}$ será inicializado aleatoriamente y se minimizará la distancia de la proyección de cada punto al centroide correspondiente, según la siguiente expresión:

$$\sum_{i=1}^{N} \left\| \phi(x_i) - \mu_{f(x_i)} \right\|^2$$

El primer paso es simplemente asignar cada punto a aquel *cluster* cuyo centroide esté más cerca. El segundo paso es recalcular el centroide de cada *cluster* teniendo en cuenta los nuevos datos asignados. El número de posibles *clusterings* es finito así que después de un número finito de iteraciones el algoritmo convergerá a un agrupamiento estable. Vamos a implementar este algoritmo de una forma dual, reprensentado el cluster como una matriz C, de dimensiones $N \times R$ definida como:

$$C_{ik} = \begin{cases} 1 & \text{si } x_i \text{ pertenece al } \textit{cluster } k \;\Rightarrow\; f(x_i) = k \\ 0 & \text{cualquier otro caso} \end{cases}$$

Decimos que el *clustering* viene dado por la matriz C, ya que cada fila contiene únicamente un 1 y la suma de las componentes de la columna k, nos da el número de puntos asignados al *cluster* k. A partir de esta matriz de *cluster* podemos obtener las coordenadas de cada centroide calculando la matriz $\Phi^T CD$, donde cada una de las R columnas contiene las coordenadas de uno de los R posibles centroides.

Para el cálculo de la matriz anterior necesitamos:

- La matriz Φ contiene los datos en el espacio de características: $\Phi = \{\phi(x_1), \cdots, \phi(x_N))\}$
- La matriz C es la matriz de *clustering*.
- La matriz diagonal D, que contiene en la diagonal principal la inversa del tamaño de cada *cluster*

La distancia a un nuevo vector $\phi(x)$ de los centroides viene dada por la siguiente expresión:

$$\begin{aligned} \|\phi(x) - \mu_k\|^2 &= \|\phi(x)\|^2 - 2\langle \phi(x), \mu_k \rangle + \|\mu_k\|^2 = \\ &= \kappa(x,x) - 2\left(\mathbf{k}^T CD\right)_k + \left(DC^T \Phi\Phi^T CD\right)_{kk} \end{aligned}$$

donde $\mathbf{k} = [k(x, x_1), ..., k(x, x_N)]^T$ es el vector que contiene el producto escalar entre $\phi(x)$ y el resto de puntos.

Por lo tanto, la asignación de cluster de $\phi(x)$ se realizará de la siguiente manera:

$$f(x) = \arg \min_{1 \leq k \leq R} \|\phi(x) - \mu_k\|^2 = \arg \min_{1 \leq k \leq R} \left(DC^T KCD\right)_{kk} - 2\left(\mathbf{k}^T AD\right)_k \tag{2.5}$$

donde K es la matriz de *kernels* de los datos.

A pesar de su popularidad, este algoritmo es propenso a caer en mínimos locales, debido a que la optimización no es convexa. Pero es un método muy ultilizado por su rapidez.

2.2.3. Solución relajada: métodos espectrales

En esta sección se va a explicar una relajación en el problema anterior para obtener una aproximación convexa.

Vamos a partir del cálculo de los centroides realizado en el algoritmo K-medias, siendo las coordenadas de μ_k una de las R columnas de la matriz resultante de la operación $\Phi^T CD$, donde Φ es la matriz de datos en el espacio de características, C es la matriz de asignación de puntos a cada *cluster* y D es la matriz diagonal con la inversa del tamaño de cada *cluster* en la diagonal.

Si consideramos ahora la matriz $\Phi^T CDC^T$, obtendremos N columnas con copias de las coordenadas de los centroides correspondientes a cada uno de los N datos de entrada. Ahora podemos calcular la distancia (al cuadrado) de cada dato a su correspondiente centroide según la siguiente expresión:

$$\begin{aligned} \left\|\Phi^T CDC^T - \Phi^T\right\|_F^2 &= \left\|\left(I_N - CDC^T\right)\Phi^T\right\|_F^2 \\ &= tr\left(\Phi^T\left(I_N - CDC^T\right)\Phi\right) \\ &= tr\left(\Phi\Phi^T\right) - tr\left(\sqrt{D}C^T\Phi\Phi^T C\sqrt{D}\right) \end{aligned}$$

Por tanto podemos definir la funcion de coste del agrupamiento como:

$$ss(A) = tr\,(K) - tr\left(\sqrt{D}C^T KC\sqrt{D}\right) \tag{2.6}$$

donde K es la matriz de *kernel* y D es la matriz diagonal con la inversa de la suma de las columnas de C. Para minimizar el coste del agrupamiento, $ss\,(C)$, tenemos que maximizar el segundo término de la expresión de dicho coste:

$$\text{máx}_C \quad tr\left(\sqrt{D}C^T KC\sqrt{D}\right) \tag{2.7}$$

Este criterio demuestra que un agrupamiento óptimo es invariante a traslaciones de los ejes de coordenadas, y será cierto para cualquier suma de distancias al cuadrado para cualquier punto fijado.

Nuestro objetivo ahora es relajar el problema para poder conseguir un problema convexo, pero cuya solución no se corresponderá precisamente con un clustering. Nótese que las matrices C y D satisfacen:

$$\sqrt{D}C^T C\sqrt{D} = I_R = H^T H \tag{2.8}$$

donde $H = C\sqrt{D}$ y el único requisito es que esta matriz H sea ortonormal, de dimensión $N \times R$, satisfaga la segunda igualdad de la ecuación (2.8).

Por tanto, la maximización relajada para el clustering se obtiene resolviendo la siguiente expresión:

$$\max_{H} \quad tr\left(H^T K H\right) \tag{2.9}$$

sujeto a $H^T H = I_R$.

Como hemos anticipado, la solución obtenida no se corresponde con un agrupamiento directamente, pero puede guiar para conseguirlo.

El máximo de la expresión $tr\left(H^T K H\right)$ sobre todas las matrices de dimensión $N \times R$ que satisfacen que $H^T H = I_R$ es igual a la suma de los primeros R autovalores de la matriz K:

$$\max_{H^T H = I_N} \quad tr\left(H^T K H\right) = \sum_{k=1}^{R} \lambda_k \tag{2.10}$$

donde λ_k es el k-ésimo autovalor de la matriz K y se cumple que $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$

La solución óptima viene dada por la expresión:

$$H = V_R Q \tag{2.11}$$

donde Q es una matriz arbitraria y ortonormal, de dimensión $R \times R$ y V_R es una matriz de dimensión $N \times R$ compuesta por los primeros R autovectores de K. Además podemos acotar el error del mejor clustering:

$$\begin{aligned} \min_{C \text{ matriz de clustering}} ss\left(C\right) &= tr\left(K\right) - \max_{C \text{ matriz de clustering}} tr\left(\sqrt{D} C^T K C \sqrt{D}\right) \\ &= tr\left(K\right) - \max_{H^T H = I_R} \quad tr\left(H^T K H\right) = \sum_{k=R+1}^{N} \lambda_k \end{aligned}$$

En resumen, la matriz de clustering de los datos la obtendremos a partir de la rotación de los autovectores de K según una transformación ortonormal.

2.3. Cortes Normalizados

Este algoritmo transforma la tarea de *clustering* en un problema de partición de grafos basado en el concepto de *corte normalizado* [8] para resolverlo. Dicho criterio mide tanto la disimilitud entre diferentes grupos como la similitud entre elementos del mismo grupo. La optimización del criterio *corte*

normalizado se resuelve con buena eficiencia computacional como un problema generalizado de autovalores.

Esta propuesta parte de la formulación teórica de un agrupamiento basado en grafos. Según esta propuesta, un conjunto de puntos en un espacio de características arbitrario, X, es representado como un grafo ponderado no dirigido G, donde dichos puntos son los nodos del grafo y existe una arista entre cada par de nodos con una ponderación, $w(i,j)$, está definida por una función de similitud entre los nodos x_i y x_j. Para realizar el agrupamiento, buscaremos una partición del conjunto de nodos en conjuntos disjuntos $E_1, E_2, \ldots, E_m$, donde la medida de similitud entre nodos en el conjunto E_i sea alta y entre nodos de diferentes grupos E_i, E_j sea baja.

2.3.1. Agrupamiento como partición de un grafo

Un grafo G puede ser dividido en dos conjuntos disjuntos,A y B, donde la unión de los nodos de A y B contienen a todos los nodos del grafo y entre ambas particiones no comparten ningún nodo, mediante la eliminación de las aristas que conectan las dos partes. El grado de disimilitud entre estos dos grupos puede ser calculado como el peso total de las aristas que han sido eliminadas.

En teoría de grafos esto es denominado *corte*:

$$corte\,(A,B) = \sum_{u \in A, v \in B} w\,(u,v)\,. \tag{2.12}$$

La partición óptima del grafo es aquella que minimiza el valor de *corte* (expresión 2.12). Aunque existe un número exponencial de posibles particiones, el problema de resolver el mínimo se puede resolver con algoritmos eficientes.

Sin embargo, este criterio no es totalmente válido para realizar una correcta partición debido a que favorece a las particiones de pequeños grupos aislados de nodos en un grafo, ya que en muchos casos el corte mínimo es aquél en que la partición consta de un solo nodo.

Para evitar este problema, se propone una nueva medida de disimilitud entre dos grupos. En vez de evaluar el peso total de las aristas que conectan las dos particiones, vamos a calcular el coste de *corte* como la fracción entre el número total de conexiones de todos los nodos del grafo. Llamaremos a esta medida el *corte normalizado* y se calcula de la siguiente manera:

$$Ncorte\,(A,B) = \frac{corte\,(A,B)}{asoc\,(A,X)} + \frac{corte\,(A,B)}{asoc\,(B,X)}$$

donde $asoc\,(A,B) = \sum_{a\in A, t\in X} w\,(u,t)$ es el número total de conexiones entre los nodos de A y todos los nodos del grafo. Con esta definición de disociación entre los grupos, el corte que divide los grupos en nodos aislados ya no le corresponde un valor pequeño de *Ncorte*, ya que el valor del corte será un gran porcentaje de las conexiones totales desde el pequeño grupo al resto de nodos del grafo.

En la misma línea, podemos definir una medida total de la asociación normalizada dentro de los grupos para una partición dada de la siguiente manera:

$$Nasoc\,(A,B) = \frac{asoc\,(A,A)}{asoc\,(A,X)} + \frac{asoc\,(B,B)}{asoc\,(B,X)} \tag{2.13}$$

donde $asoc\,(A,A)$ y $asoc\,(B,B)$ son el número total de pesos de los vértices que conectan los nodos pertenecientes a A y B respectivamente.

Otra propiedad importante de esta definición de asociación y disociación de una partición es que las expresiones anteriores están relacionadas de la siguiente forma:

$$Ncorte\,(A,B) = \frac{asoc\,(A,A)}{asoc\,(A,X)} + \frac{asoc\,(B,B)}{asoc\,(B,X)} = 2 - Nasoc\,(A,B) \tag{2.14}$$

Por lo tanto, aplicando este criterio, minimizar la disociación entre los grupos y maximizar la asociación dentro de cada grupo es equivalente y ambas acciones se satisfacen simultáneamente.

2.3.2. Cálculo de la partición óptima

Dada una partición en dos grupos de nodos, A y B pertenecientes a un grafo, G, decimos que t es un vector indicador de dimensión $N = |X|$, siendo $t_i = 1$ si el nodo i pertenece al grupo A y $t_i = -1$ en caso contrario. Denominamos $d\,(i) = \sum_j w\,(i,j)$ al número total de conexiones desde el nodo i al resto de nodos. Con la definiciones anteriores de x y d, podemos reescribir la expresion de $Ncorte\,(A,B)$ como:

$$\begin{aligned} Ncorte\,(A,B) &= \frac{asoc\,(A,A)}{asoc\,(A,X)} + \frac{asoc\,(B,B)}{asoc\,(B,X)} \\ &= \frac{\sum_{(t_i>0,t_j<0)} -w_{ij}t_it_j}{\sum_{t_i>0} d_i} + \frac{\sum_{(t_i<0,t_j>0)} -w_{ij}t_it_j}{\sum_{t_i<0} d_i} \end{aligned} \quad (2.15)$$

Dadas la matriz diagonal D de dimensión $N \times N$, con d en su diagonal, la matriz simétrica W de dimensión $N \times N$ con $W\,(i,j) = w_{ij}$, $k = \frac{\sum_{t_i>0} d_i}{\sum_i d_i}$ y $\mathbf{1}$ es el vector de unos de dimensión $N \times 1$ podemos reescribir la expresión (2.15) de la siguiente forma:

$$Ncorte\,(A,B) = \frac{(1+t)^T\,(D-W)\,(1+t)}{k1^TD1} + \frac{(1-t)^T\,(D-W)\,(1-t)}{(1-k)\,1^TD1} \quad (2.16)$$

Operando se llega a la siguiente expresión:

$$\min_t Ncorte\,(t) = min_y \frac{y^T\,(D-W)\,y}{y^TDy} \quad (2.17)$$

donde $y = (1+t) - b\,(1-t)$ y $b = \frac{k}{1-k}$ y debe cumplirse que $y\,(i) \in \{1, -b\}$ e $y^TD1 = 0$

Si la solución y es relajada para conseguir valores reales, podemos minimizar la ecuación (2.17) mediante la resolución de un problema de autovalores generalizado:

$$(D-W)\,y = \lambda Dy \quad (2.18)$$

Sin embargo, tenemos dos restricciones sobre y, la primera $y^TD1 = 0$ se satisface automáticamente con la solución del sistema generalizado de autovalores. Para demostrarlo es necesario transformar la expresión (2.18) en un sistema de autovalores típico:

$$D^{-\frac{1}{2}}\,(D-W)\,D^{-\frac{1}{2}}z = \lambda z \quad (2.19)$$

donde $z = D^{\frac{1}{2}}y$. Se verifica que $z_0 = D^{\frac{1}{2}}\mathbf{1}$ es un autovector de la matriz $D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}}$ con autovalor igual a cero. Además, dicha matriz es simetrica y semidefinida positiva ya que $(D-W)$ es semidefinida positiva. Por tanto, z_0 es el autovector cuyo autovalor es el más pequeño del sistema (2.19) y todos los autovectores de (2.19) son perpendiculares entre sí. En particular, z_1, el segundo autovector con autovalor asociado más pequeño, es perpendicular a z_0. Trasladando estas conclusiones a la expresión (2.18), tenemos que $y_0 = \mathbf{1}$ es el autovector con autovalor asociado más pequeño e igual a 0 y que $0 = z_1^T z_0 = y_1^T D\mathbf{1}$, donde y_1 es el segundo autovector con autovalor más pequeño de (2.18).

Por lo tanto, el segundo autovector con autovalor más pequeño del sistema de autovalores generalizado (2.18) es la solución aproximada de nuestro problema de corte normalizado. La única razon por la que no es necesariamente la solución de nuestro problema original es porque la segunda restricción de y es que $y(i)$ debe tomar 2 posibles valores discretos, $y(i) \in \{1, -b\}$, y con la solución anterior no se cumple. Los valores de los que está compuesto el autovector son continuos, por tanto debemos determinar un punto de ruptura o *splitting point* a partir del cual discriminar si los puntos pertenecen a un grupo u a otro de la partición.

Existen diferentes formas de seleccionar un *splitting point*:

- Asignarle al *splitting point* el valor 0.
- Calcular la media de los valores de las componentes del autovector y asignarle ese valor.
- Calcular la mediana de los valores de las componentes del autovector y asignarle ese valor.
- Seleccionar el *splitting point* que mejor corte normalizado genere.

2.3.3. Algoritmo de Agrupamiento

El algoritmo de agrupamiento consiste en dados unos datos de entrada, X, transformarlos en un grafo ponderado G en el que los pesos de cada arista que conecta dos nodos es la medida de similitud entre esos dos nodos. A continuación resolvemos el siguiente problema de autovectores $(D-W)y = \lambda Dy$, obteniendo los autovectores con autovalor asociado más pequeño. Tomamos el autovector cuyo autovalor asociado sea el segundo más pequeño. Por último, debemos decidir si la particion a la que hemos llegado es válida o hemos de seguir dividiendo recursivamente.

En resumen, el algoritmo de agrupamiento sigue los siguientes pasos:

1. Construir el grafo G a partir de los datos de entrada, X.
2. Resolver problema de autovalores de $(D - W)\, y = \lambda D y$.
3. Coger el autovector y asociado al segundo autovalor más pequeño.
4. Elegir *splitting point* que marque la división de los puntos en dos grupos según el valor de su coordenada en el autovector.
5. Decidir si hay que seguir con la división, en cuyo caso para cada una de las dos particiones se vuelve al punto 2.

2.4. Clustering Espectral. K-medias ponderado

El objetivo de este método [2], como el de los anteriores, es conseguir agrupar los datos de entrada de tal forma que los datos dentro de cada grupo sean similares y sin embargo los datos entre grupos sean distintos.

Como en otros métodos, éste se basa en conceptos anteriormente desarrollados como matriz de similitud y cortes normalizados, pero va más alla proponiendo dos funciones de coste que permiten evaluar la bondad del *clustering* y por tanto para conseguir cumplir el objetivo de encontrar un agrupamiento eficiente.

2.4.1. Cortes normalizados

Este método parte del criterio del corte normalizado de Shi y Malik [8], en el que para dos subconjuntos A y B visto en el apartado anterior. Para este método se extiende el corte normalizado a R particiones, según la siguiente expresión:

$$Ncorte\,(E, W) = \sum_{r=1}^{R} \frac{e_r^T\,(D - W)\,e_r}{e_r^T D e_r} \tag{2.20}$$

donde $E = (e_1, ..., e_R) \in \Re^{P \times R}$ representa la partición de los datos, siendo el vector e_r el indicador en $\Re^P$ del r-ésimo *cluster* , tal que, $e_r \in \{0, 1\}^P$ es tal que tiene una componente no nula sólo en los puntos pertenecientes a dicho cluster.La matriz D denota la matriz diagonal cuyo elemento i-ésimo es la

suma de los elementos de la i-ésima fila de W, por lo que, $D = \text{Diag}\,(W\mathbf{1})$, donde $\mathbf{1}$ es el vector en $\Re^P$ compuesto de unos.

2.4.2. Relajación espectral

Extendiendo el resultado de Shi y Malik para dos *clusters* se llega a la siguiente relajación espectral:

- **Proposición 1**: Para la partición E en R *clusters*, el corte normalizado es igual a:

$$Ncorte\,(W, E) = R - \text{tr}\left(S^T D^{-1/2} W D^{-1/2} S\right) \tag{2.21}$$

  para cualquier matriz $S \in \Re^{P\times R}$ tal que:

  1. Las columnas de $D^{-1/2}S$ sean contantes a trozos con respecto a los *clusters* E
  2. Y tenga las columnas ortonormales ($S^T S = I$)

- **Proposición 2**: El máximo de $\text{tr}\left(S^T D^{-1/2} W D^{-1/2} S\right)$ sobre matrices $S \in \Re^{P\times R}$ tal que $S^T S = I$ es la suma de los R mayores autovalores de $D^{-1/2}WD^{-1/2}$. Esto afecta a todas la matrices S de la forma $S = YB_1$, donde $Y \in \Re^{P\times R}$ es cualquier base ortonormal del R-ésimo subespacio principal de $D^{-1/2}WD^{-1/2}$ (Y contiene los autovectores de $D^{-1/2}WD^{-1/2}$) y B_1 es una matriz ortogonal arbitraria en $\Re^{R\times R}$

2.4.3. Rounding

El procedimiento de *rounding* está basado en la minimización de una métrica entre la solución relajada y el conjunto completo de soluciones discretas permitidas. Se van a tener en cuenta dos métricas diferentes.

Comparación de subespacios

Las soluciones del problema relajado son definidas como la matriz ortogonal, $S_{eig} = YB_1$, donde $Y \in \Re^{P\times R}$ es cualquier base ortonormal del R-ésimo

subespacio principal de M y B_1 es una matriz ortonormal arbitraria. El conjunto de matrices E correspondiente a la partición E y que satisface las restricciones (1) y (2) son de la forma:

$$S_{part} = D^{1/2} E \left(E^T D E \right)^{-1/2} B_2 \tag{2.22}$$

donde B_2 es una matriz ortogonal arbitraria. Ya que ambas matrices son definidas como matrices ortogonales, tiene sentido comparar subespacios definidos por sus columnas. Una forma usual de realizarlo es comparar la proyeccion ortogonal de dichos subespacios, es decir, calcular la norma Forbenius entre $S_{eig} S_{eig}^T = Y Y^T$ y la proyección ortogonal $\Pi_0 (W, E)$ sobre el subespacio definido por las columnas de $D^{1/2} E = D^{1/2} (e_1, ..., e_r)$, de la siguiente manera:

$$\Pi_0 (W, E) = S_{part} S_{part}^T = D^{1/2} E \left(E^T D E \right)^{-1} E^T D^{1/2} \tag{2.23}$$

Por tanto, la función de coste queda definida de la siguiente forma:

$$J_1 (W, E) = \frac{1}{2} \left\| Y (W) Y (W)^T - \Pi_0 (W, E) \right\|_F^2 \tag{2.24}$$

Sabiendo que $Y (W) Y (W)^T$ y $\Pi_0 (W, E)$ son proyecciones ortogonales sobre subespacios lineales de dimensión R, se llega a la siguiente expresión:

$$J_1 (W, E) = R - \sum_r \frac{e_r^T D^{1/2} Y (W) Y (W)^T D^{1/2} e_r}{e_r^T D e_r} \tag{2.25}$$

Normalización de autovectores

Se define otra función de coste alternativa mediante la eliminación del escalado introducido por D, por tanto, multiplicaremos Y por $D^{-1/2}$ y reortogonalizaremos para obtener $U = D^{1/2} Y \left(Y^T D^{-1} Y \right)^{-1/2}$, donde se usa cualquier de matriz raiz de $Y^T D^{-1} Y$. Hay que notar que esto es equivalente a considerar un problema de autovalores generalizado.

Por tanto, la función de coste tiene la siguiente expresión:

$$J_2 (W, E) = \frac{1}{2} \left\| U (W) U (W)^T - E \left(E^T E \right)^{-1} E^T \right\|_F^2 \tag{2.26}$$

Las dos funciones de coste se caracterizan por la habilidad de la matriz W para producir la partición E cuando se usan sus autovectores. Minimizando con respecto a E se llega a nuevos algoritmos de clustering. Minimizando respecto a la matriz W, para una particion dada, se llega a algoritmos para aprendizaje de la matriz de similitud.

2.4.4. Algoritmos de clustering espectral

Una vez hemos descrito las dos funciones de coste que van a ser utilizadas, utilizaremos el algoritmo de K-medias como método para la minimización de dichas funciones. Para ello, resulta muy útil expresar las funciones de coste como una medida ponderada de la distorsión:

- **Función de coste 1**

  Dada la matriz de similitud W y la matriz $Y = (y_1, ..., y_N)^T$, donde $y_i \in \Re^R$ es una base ortonormal del R-ésimo subespacio principal de $D^{-1/2}WD^{-1/2}$ y $d_i = D_{ii}$ para todo x_i, se puede expresar la primera función de coste como:

$$J_1(W, E) = \min_{(\mu_1,...,\mu_R)\in\Re^{R\times R}} \sum_r \sum_{x_i\in C_r} d_i \left\| y_i d_i^{-1/2} - \mu_r \right\|^2 \tag{2.27}$$

  donde μ_r es el centroide del *cluster* r y C es la matriz de *clustering*, por tanto C_r la columna de esa matriz correspondiente al *cluster* r.

  Para realizar la minimización de la función de coste se aplica el siguiente algoritmo:

  1. Obtener la matriz de similitud $W \in \Re^{N\times N}$ a partir de los datos de entrada X.
  2. Calcular los R primeros autovectores de $D^{-1/2}WD^{-1/2}$ donde $D = \text{diag}(W\mathbf{1})$.
  3. Formamos la matriz Y con esos R primeros autovctores: $Y = (y_1, ..., y_N)^T \in \Re^{N\times R}$ y $d_i = D_{ii}$.
  4. Inicializar la partición, por tanto asignación inicial de los puntos a un cluster. Creación matriz de *clustering* C.
  5. Aplicar algoritmo K-medias ponderado mientras que la partición A no sea estacionaria,

     a) Para todo r, $\mu_r = \frac{\sum_{x_i\in C_r} d_i^{1/2} y_i}{\sum_{x_i\in C_r} d_i}$

b) Para todo x_i, se asigna x_i a C_r, donde $r = \text{argmin}_{r'} \left\| y_i d_i^{-1/2} - \mu_{r'} \right\|$

6. Como resultado, el algoritmo proporciona la partición C y la medida de distorsión: $\sum_r \sum_{x_i \in C_r} d_i \left\| y_i d_i^{-1/2} - \mu_r \right\|^2$

- **Función de coste 2**

  Dada la matriz de similitud W y la matriz Y, base ortonormal del R-ésimo subespecio principal de $D^{-1/2}WD^{-1/2}$, y $U = D^{1/2}Y\left(Y^TDY\right)^{-1/2}$, se puede expresar la segunda función de coste como:

$$J_2(W, E) = \min_{(\mu_1,...,\mu_R)\in\Re^{R\times R}} \sum_r \sum_{x_i \in C_r} \|u_p - \mu_r\|^2 \tag{2.28}$$

  Para realizar la minimización de la función de coste se aplica el siguiente algoritmo:

1. Obtener la matriz de similitud $W \in \Re^{N\times N}$ a partir de los datos de entrada X.
2. Calcular los R primeros autovectores de $D^{-1/2}WD^{-1/2}$ donde $D = \text{diag}(W\mathbf{1})$.
3. Formamos la matriz Y con esos R primeros autovectores: $Y = (y_1, ..., y_N)^T \in \Re^{N\times R}$ y $d_i = D_{ii}$.
4. Calcular $U = D^{1/2}Y\left(Y^TDY\right)^{-1/2}$.
5. Tenemos $U = (u_1, ..., u_N)^T \in \Re^{N\times R}$
6. Inicializar la partición, por tanto asignación inicial de los puntos a un cluster. Creación matriz de *clustering* C.
7. Aplicar algoritmo K-medias ponderado mientras que la partición A no sea estacionaria,
   a) Para todo r, $\mu_r = \frac{1}{|C_i|}\sum_{x_i\in C_r} u_i$.
   b) Para todo x_i, se asigna x_i a C_r, donde $r = \text{argmin}_{r'} \|u_i - \mu_{r'}\|$
8. Como resultado, el algoritmo proporciona la partición A y la medida de distorsión: $\sum_r \sum_{x_i\in C_r} \|u_i - \mu_r\|^2$

2.5. Clustering Espectral Refinado

El algoritmo de *clustering* espectral refinado [9] toma como partida el algoritmo desarrollado por Ng, Jordan y Weiss [5]. A continuación se muestra

este algoritmo de partida para posteriormente desarrollar el algoritmo de *clustering* espectral refinado.

2.5.1. Algoritmo de Ng-Jordan-Weiss

El algoritmo de Ng-Jordan-Weiss(NJW) es el punto de partida para el algoritmo de clustering espectal refinado y consta de las siguientes fases:

- Dado un conjunto de N puntos $X = \{x_1, ..., x_N\}$, para realizar el agrupamiento necesitamos calcular la matriz de afinidad o similitud W en función de una medida de similitud entre cada par de puntos. A partir de la matriz W, calculamos la matriz diagonal D, que contiene en cada elemento de la diagonal la suma de las afinidades de cada punto con el resto.
- Una vez tenemos W y D calculamos la matriz del Laplaciano normalizado L, según la expresión: $L = D^{-1/2}WD^{-1/2}$ y sus autovectores correspondientes. Una vez seleccionado manualmente el numero de clusters, R, tomamos los R autovectores cuyo autovalor asociado sea mayor y formamos con ellos la matriz $Y = [y_1, ..., y_R]$.
- Calculamos la matriz Y_{norm} mediante la normalización de las filas de la matriz Y. A continuación tratamos cada fila de Y_{norm} como un punto en $\Re^C$, agrupamos con K-medias y asignamos cada punto original, x_i, al *cluster* correspondiente, c, si y sólo si la fila i de la matriz Y_{norm} fue asignada al *cluster* c.

En resumen, los pasos a seguir por este algoritmo son los siguientes:

1. Conjunto de N puntos: $X = \{x_1, ..., x_N\}$ en $\Re^l$
2. Matriz de similitud o afinidad W, donde:
   $W_{ij} = w(i, j)$ es la similitud entre el punto x_i y x_j $W_{ii} = 0$
3. $D_{ii} = \sum_{j=1}^{n} W_{ij}$
4. $L = D^{-1/2}WD^{-1/2}$
5. Seleccionar el número de clusters, R
6. Autovectores de L: $Y = [y_1, ..., y_R] \in \Re^{N \times R}$

7. $Y_{norm\ ij} = \frac{Y_{ij}}{\left(\sum_j Y_{ij}^2\right)^{1/2}}, Y_{norm} \in \Re^{N \times R}$

8. Agrupar cada fila de Y_{norm} vía K-medias

2.5.2. Estimación del número de *clusters*

El número de clusters suele ser un parámetro introducido manualmente y no se han realizado muchas investigaciones sobre cómo poder seleccionarlo automáticamente. A continuación se proponen dos posibles soluciones, una basada en el análisis de los autovalores de la matriz L y otra basada en los autovectores de la misma matriz.

Una posible aproximación para descubrir el número de grupos es analizar los autovalores de la matriz L. El primer autovalor (autovalor de mayor magnitud e igual a 1) se repetirá con multiplicidad igual al numero de grupos, R. Esto implica que podremos estimar R contando el número de autovalores iguales a 1. Cuando los datos son perfectamente separables podemos comprobar que la multiplicidad del autovalor mayor es igual al número de grupos. Sin embargo, si los grupos no están claramente separados los valores de dichos autovalores empiezan a desviarse de 1, lo que lleva a que el criterio de elección sea más delicado y complicado.

Una aproximación alternativa sería buscar una caída en la magnitud de los autovalores, pero este proceso no es del todo exacto. Los autovalores de la matriz L son la unión de los autovalores de las submatrices correspondientes a cada *cluster*. Esto implica que los autovalores dependen de la estructura de cada *cluster* invidualmente por lo que no se pueden hacer suposiciones sobre sus valores. En particular, el salto entre el autovalor correspondiente al cluster R y el siguiente podría ser pequeño o grande.

Una alternativa al método anterior para la estimación del número de *clusters* es analizar los autovectores de la matriz L. Después de colocar la matriz L según los *clusters*, en el caso ideal (cuando L es estrictamente una matriz diagonal por bloques, con bloques $L^{(r)}$, con $(r = 1, ..., R)$ sus autovalores y autovectores son la unión de los autovalores y autovectores de cada bloque, rellenando apropiadamente con ceros. A medida que los autovalores de los bloques son diferentes, cada autovector será distinto de cero sólo en las entradas correspondientes a un único bloque.

$$\hat{Y} = \begin{bmatrix} \vec{y}^{(1)} & \vec{0} & \vec{0} \\ \vec{0} & \cdots & \vec{0} \\ \vec{0} & \vec{0} & \vec{y}^{(R)} \end{bmatrix}_{N \times R}$$

Donde $\vec{y}^{(c)}$ es el autovector de la submatriz $L^{(c)}$, correspondiente al cluster c. Como se dijo anteriormente, el autovalor 1 va a ser repetido con multiplicidad C (número de *clusters*), así que la solución puede ser más sencilla, rotando la matriz Y con los autovectores de L hasta conseguir una matriz $\hat{Y}$, donde los vectores que la forman sean ortonormales. Por tanto, Y va a ser reemplazada por $\hat{Y} = Y \cdot Q$, para cualquier matriz ortogonal Q tal que $Q \in \Re^{R \times R}$.

Como los autovectores de L son la unión de los autovectores de cada bloque individual (relleno de ceros), tomando más de los R primeros autovectores obtendremos más de una entrada distinta de cero en algunas de las filas. Tomando un número menor de autovectores no tendremos una base completa del subespacio, así que dependiendo de la matriz inicial Y, podría existir o no dicha rotación. Hay que decir que estas observaciones son independientes de la diferencia de magnitud entre los autovalores.

A partir de estas observaciones es posible predecir el número de grupos. Por cada posible número de grupos R, recuperaremos la rotación que mejor alinea las columnas de Y con el sistema de coordenadas canónico. Denotamos $Z \in \Re^{N \times R}$ a la matriz obtenida después de rotar los autovectores Y ($Z = Y \cdot Q$) y a $T_i = \max_j Z_{ij}$. Queremos recuperar la rotación Q para que en cada fila de Z sólo tengamos como mucho una entrada distinta de cero. Definimos la función de coste:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{R} \frac{Z_{ij}^2}{T_i^2}$$

Minimizando esta función de coste sobre todas las posibles rotaciones proporcionaremos un alineamiento con el sistema de coordenadas canónicas. Esto se hará usando un esquema basado en de descenso por gradiente. El número de grupos se toma como aquel que proporciona el mínimo coste. Si varios agrupamientos obtienen un coste parecido, se elegirá el de mayor número de grupos.

2.5.3. Algoritmo de *clustering* espectral refinado

El método propuesto para la estimación del número de grupos de forma automática tiene dos consecuencias:

1. Después de alinear con el sistema de coordenadas canónicas, se pueden suprimir las filas de Z que no tengan valores altos lo que eliminará el

proceso iterativo final del K-medias [3], que a menudo requiere de un alto número de iteraciones y depende mucho de su inicialización.

2. Ya que el clustering final puede realizarse mediante dicha supresión, se obtendrán resultados del clustering para todos los grupos examinados en un pequeño coste adicional.

Cuando los datos no sean claramente separables, podremos seguir utilizando K-medias para agrupar las filas de la matriz Z. Sin embargo, ya que los datos están ahora alineados con el sistema de coordenadas canónicas, se podrá obtener una excelente inicialización mediante la supresión de valores no máximos con lo que serán necesarias muy pocas iteraciones.

Algoritmo

Dado el conjunto de puntos que queremos agrupar, X, la matriz de similitud W formada por la medida de simlitud entre todos los puntos de entrada. A partir de esta matriz, calculamos la matriz diagonal D, que contiene en cada elemento de la diagonal la suma de las afinidades de cada punto con el resto. Una vez tenemos W y D, calculamos la matriz L, según la expresión: $L = D^{-1/2}WD^{-1/2}$ y sus autovectores correspondientes. Una vez seleccionado manualmente el numero de clusters, C, tomamos los R autovectores cuyo autovalor asociado sea mayor y formamos con ellos la matriz $Y = [y_1, ..., y_C]$. Recuperar la rotación Q que mejor alinea las columnas de Y con el sistema de coordenadas cartesianas usando un esquema basado en descenso por gradiente. Evaluar el coste de alineamiento para cada número de grupo hasta R, quedándonos con el número de grupo mayor, J_{best}, que menor coste de alineamiento consigue. Tomamos el alineamiento resultante, Z, de los primeros J_{best} autovectores y asignamos a cada punto original x_i al *cluster* c si y sólo si $\text{máx}_j \left(Z_{ij}^2\right) = Z_{ir}^2$. Si los datos no son claramente separables, se recomienda usar el paso previo para incializar K-medias o *EM*, agrupando las filas de Z.

En resumen, los pasos a seguir por este algoritmo son los siguientes:

1. Conjunto de datos de entrada: $X = \{x_1, ..., x_N\}$ en $\Re^l$
2. Matriz de similitud o afinidad W, donde: $W_{ij} = w(i, j)$ es la similitud entre el punto x_i y x_j $W_{ii} = 0$
3. $D_{ii} = \sum_{j=1}^{n} W_{ij}$
4. $L = D^{-1/2}WD^{-1/2}$

5. Seleccionar R
6. Autovectores de L: $Y = [y_1, ..., y_R] \in R^{N \times R}$
7. Recuperar rotacion Q
8. $J = \sum_{i=1}^{N} \sum_{j=1}^{R} \frac{Z_{ij}^2}{T_i^2} \overset{costemenor}{\rightarrow} J_{best}$
9. $\text{máx}_j \left(Z_{ij}^2 \right) = Z_{ir}^2$

Capítulo 3

Autoorganización de documentos

En todo ejercicio de minería de datos es necesario estudiar y comprender cómo son los datos de entrada y realizar transformaciones sobre los mismos para que puedan ser aplicados a los algoritmos de minería de datos, en este caso concreto a nuestros algoritmos de *clustering* espectral. Por tanto, en este capítulo estudiaremos de qué manera se parametrizan los documentos para que puedan ser tratados por los algoritmos.

Por otro lado, una vez que obtenemos un agrupamiento de los datos mediante los algoritmos de *clustering*, es necesario evaluar la calidad del mismo. En este capítulo también veremos qué medida de calidad vamos a utilizar para evaluar los clusterings proporcionado por cada algoritmo y a partir de ahí determinar cuál de ellos obtiene mejores prestaciones, objetivo de este proyecto.

3.1. Parametrización de documentos

A la hora de realizar un *clustering* con documentos es necesario llevar a cabo tareas de preprocesado y transformación de los mismos, puesto que los documentos en texto plano no son válidos para llevar a cabo operaciones algebraicas presentes en todos nuestros algoritmos de *clustering*.

El principal problema cuando trabajamos con lenguaje natural es que el contexto tiene importancia en el significado del texto. La parametrización de los documentos [4] consiste en transformar el texto en un conjunto de bloques,

proceso denominado indexación de términos. Existen diferentes niveles de indexación, que depende del grado de descomposición del documento:

- **Descomposición de palabras.**

  Los bloques de indexación son formados a partir de la descomposición morfológica de las palabras que conforman el documento. La técnica más popular es *n-Gram*, que genera bloques de cadenas de n caracteres. Por ejemplo: 3-Gram de la palabra *book* es *_bo*, *boo*, *ook* y *ok_*. De esta manera estamos representando modelos de similitud entre palabras. Una ventaja de esta representación es que proporciona más robustez frente errores de escritura.

- **Descomposición a nivel de palabra.**

  Se divide el texto en las palabras de las que está compuesto. El documento se transforma en un vector denominado *bag of words*, que contiene la repetitividad de las palabras que forman el texto.

- **Descomposición en grupos de palabras.**

  Se divide el texto en bloques formados por varias palabras, teniendo en cuenta su información sintáctica. Se estudia la frecuencia de esa formación de palabras en el documento.

- **Análisis semántico del documento.**

  Se analiza el significado del texto. Esta tarea es complicada puesto que no es posible extraer esa información de manera automática.

Cuando surgen ambigüedades en alguno de los niveles escogidos, se resulven haciendo uso de un nivel superior.

El nivel de análisis elegido para llevar a cabo la transformación de los documentos para este proyecto es la indexación a nivel de palabra. Como hemos comentado, previamente al clustering es necesario transformar os documentos en *bag of words*, que representa la repetitividad de las palabras en el texto. Por tanto no almacenamos palabras, sino que por cada documento registramos la fecuencia de aparición de las palabras.

Claramente esta transformación lleva consigo una pérdida de información contextual. Sin embargo, niveles de representación más sofisticados, distintos a la descomposición a nivel de palabra, no muestran mejoras consistentes y sustanciales. Además la representación *bag of words* mantiene un buen compromiso entre la expresividad y la complejidad del modelo de representación y análisis.

Se denomina *term frequency*, $TF(w, d)$, a la repetitividad de la palabra w en el documento d. Este es el valor que se calcula para cada palabra del documento que es registrado en la posición del vector que le corresponda a la palabra.

Es interesante determinar la importancia de una palabra dentro del documento que la contiene. Para ello se define el término $TFIDF$ (Term Frequency Inverse Document Frequency):

$$TFIDF(w, d) = TF(w, d) \cdot log\left(\frac{N}{DF(w)}\right) \tag{3.1}$$

donde:

- $TF(w, d)$(*Term frequency*): La frecuencia de la palabra w en un documento d.
- $DF(w)$(*Document frequency*): Número de documentos que contienen la palabra w.
- N: Número total de documentos.

La representación de documentos mediante *bag of words* suele ser menos fiable para lenguajes que provienen del latín, debido a la complejidad de construcciones y la mayor existencia de dobles sentidos e interpretaciones. Por esta razón, tanto en este proyecto como en el tratamiento de documentos en disciplinas de clustering, clasificación y *information retrieval* el idioma predominante que se utiliza es el inglés.

Previamente a la creación del *bag of words* se realiza un preprocesado lingüístico de las palabras, primero eliminando todas aquellas palabras que no aportan contenido semántico al texto (pronombres, conjunciones, artículos, etc) y de las palabras que quedan se aplican métodos de *stemming* para reducir las palabras a su raíz (o *stem*, en inglés), puesto que variaciones morfológicas de las palabras tienen la misma o similar interpretación semántica. Así por ejemplo las palabras *computer* y *computation* serán modificadas a *comput*.

3.2. Medida de similitud entre documentos

A la hora de realizar la comparación entre los documentos necesitamos definir una medida de similitud, en este caso adecuada para nuestros docu-

mentos en formato *bag of words*. Una medida de similitud debe ser alta cuanto más parecidos sean los elementos que estemos comparando, es decir dos documentos parecidos, que contengan palabras parecidas tendrán un valor de medida de similitud alto, mientras que dos documentos que no compartan apenas palabras en sus *bag of words* deberán tener una medida de similitud pequeña.

La medida de similitud escogida para la tarea de autoorganización de documentos es el coseno entre dos *bag of words*:

$$\cos(x, y) = \frac{x^T y}{\|x\| \, \|y\|} \tag{3.2}$$

donde x e y son las *bag of words* entre las que estamos midiendo la similitud. De esta manera, los valores de similitud estarán comprendidos entre 0 y 1.

3.3. Evaluación del *clustering*

Una vez tenemos los documentos agrupados en *clusters*, es necesario evaluar y juzgar la bondad de dicho *clustering*. Se va a calcular la información mutua [6] aportada por el *clustering* en función de cuanto se parece o difiere ésta del agrupamiento real de los documentos. En nuestra base datos, además de las *bag of words* de cada documento, se nos aporta la clasificación real de los mismos, es decir que cada documento tenía asociado una etiqueta o temática, dentro de unas etiquetas definidas.

Puesto que el número de categorías aportadas es fijo, nos encontramos con un problema a la hora de evaluar nuestro *clustering* y es que en nuestro caso vamos a realizar simulaciones variando el número de *clusters* (de 2 a 50). De ahí, que la información mutua sea una buena medida de bondad, puesto que mide la distancia entre los dos agrupamientos, con R y C grupos respectivamente, según la siguiente expresión:

$$MI(R, C) = \sum_{r=1}^{R} \sum_{c=1}^{C} p(R = r, C = c) log_2 \left(\frac{p(R = r, C = c)}{p(R = r) p(C = c)} \right) \; bits \tag{3.3}$$

La medida de la información mutua de nuestro *clustering* con respecto a la permutación óptima de los datos nos cuantifica sobre cuánta información puedo obtener del agrupamiento original a partir del realizado por nuestros algoritmos de *clustering*. Si el agrupamiento calculado no se parece al original,

el valor de la información mutua será cero. Si nos vamos al caso contrario, en el que conseguimos llegar a un agrupamiento exactamente igual al original, el valor de la información mutua alcanza su máximo valor, que coincide con el de la entropía [6] de la partición original, cuya expresión es:

$$I(R) = \sum_{r=1}^{R} p(R = r) log_2 \left(\frac{1}{p(R = r)} \right) \ bits \tag{3.4}$$

Por tanto, el objetivo de nuestro trabajo experimental será calcular la información mutua de los agrupamientos calculados por cada método de *clustering* espectral

Además del cálculo de la información mutua, vamos a realizar otra medida de evaluación de los resultados. Ya que disponemos de las etiquetas reales de los documentos, vamos a calcular la tasa de acierto de clasificación de los documentos a partir de los *clusterings* realizados. Una vez que tengamos el *clustering* de nuestros documentos, etiquetaremos cada uno de los grupos con la etiqueta mayoritaria de los documentos que lo componen. Posteriormente compararemos la etiqueta real con la obtenida realizando el paso anterior y veremos los niveles de tasa de acierto de cada algoritmo de *clustering* espectral.

Capítulo 4

Resultados experimentales

4.1. Descripción de las bases de datos

Una vez descritos los algoritmos de *clustering* espectral pasamos a evaluar cuál de ellos consigue mejores resultados en la tarea de autoorganización de documentos. Para ello es necesario seleccionar primero unas bases de datos adecuadas para este proceso. Se han realizado simulaciones para tres bases de datos: *hitech*, *k1b* y *ohscal*. Cada una de ellas posee características diferentes con respecto a las demás que genera unos resultados experimentales diferentes que estudiaremos en los siguientes apartados.

Las bases de datos han sido obtenidas de CLUTO [1], aplicación que trabaja también con *bag of words*. Cada una de las bases de datos está formada por tres tipos de ficheros:

- Archivo *.mat: es una matriz que representa las *bag of words* de cada documento, por tanto la fila i representa el documento i y en cada columna j se almacena la repetitividad de la palabra j del diccionario.
- Archivo *.rclass: contiene las etiquetas asignadas a cada documento, por tanto el clustering real con el que compararemos nuestros resultados.
- Archivo *.clabel: contiene las palabras del diccionario que componen los documentos de la base de datos.

Para hacernos una idea del tamaño de cada base de datos, en la siguiente tabla podemos observar el número de documentos de los que está compuesta

cada base de datos, el tamaño del diccionario y el número de palabras por documento, en media, que tiene cada base de datos.

BBDD	**num docs**	**tamaño dicc**	**num palabras/doc**
hitech	2301	22498	239
k1b	2340	21839	227
ohscal	2500	9537	104

Cuadro 4.1: Caracterización de las bases de datos de documentos

A continuación mostramos los histogramas de distribución de los documentos de cada base de datos según el número de palabras que contienen los documentos.

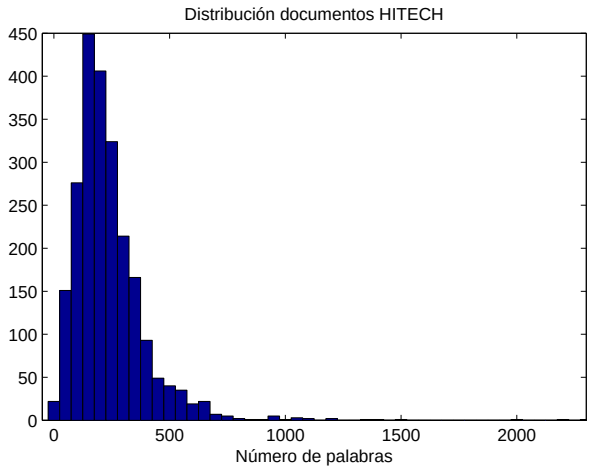

Figura 4.1: Distribución de documentos de *hitech* según el número de palabras

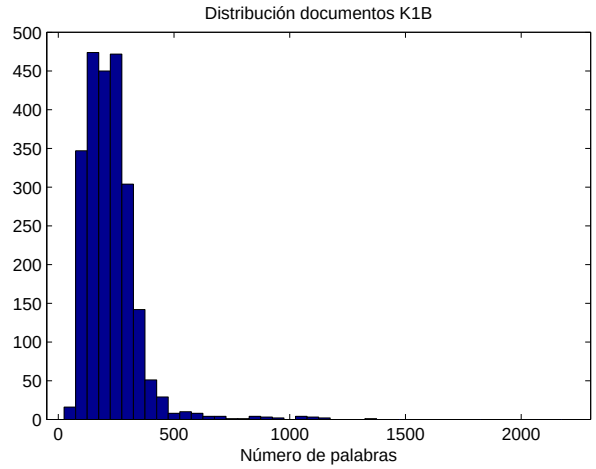

Figura 4.2: Distribución de documentos de *k1b* según el número de palabras

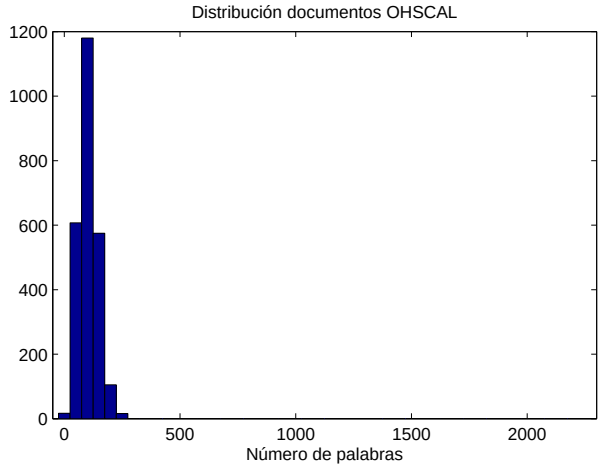

Figura 4.3: Distribución de documentos de *ohscal* según el número de palabras

Podemos observar que la distribución de los documentos de la base de datos *hitech* es menos uniforme en cuanto al número de palabras por documento que el resto de bases de datos. Concretamente esta base datos tiene documentos con 7 palabras y por otro lado documentos de hasta 2297 palabras. Ésto trae como consecuencia que la comparación entre documentos puede que sea con mucho detalle para algunos documentos y para otros resulte poco fina.

Las temáticas o etiquetas de los documentos de las bases de datos, que se viene detallado en el archivo *.rclass son las siguientes, detallamos además

el número de documentos por etiqueta:

- Base de datos *hitech*: *computers*(485), *electronics*(116), *health*(603), *medical*(429), *research*(481) y *technology*(187). Los documentos de esta base de datos tienen relación con campos de tecnología e informática o con la medicina y salud. Hay que tener cuenta que muchos documentos pueden relacionar ambos campos, puesto que en el mundo de la medicina y la salud la tecnología y aparatos electrónicos son muy importantes.
- Base de datos *k1b*: *business*(142), *entertainment*(1389), *health*(494), *politics*(114), *sports*(141) y *tech*(60). Esta base de datos tiene temáticas más diferenciadas entre sí y por tanto no existirá mucha relación entre cada uno de ellos. Un aspecto muy importante de esta base de datos es que casi el 60 % de los documentos que pertenecen a la etiqueta *entertainment*, es decir cuando validemos los resultados obtenidos con la etiqueta real, será más probable acertar etiquetando los documentos con esta etiqueta mayoritaria.
- Base de datos *ohscal*: *Antibodies*(284), *Carcinoma*(164), *DNA*(186), *In-Vitro*(201), *Molecular-Sequence-Data*(182), *Pregnancy*(357), *Prognosis*(199), *Receptors*(301), *Risk-Factors*(324), *Tomography*(302). Esta base de datos está centrada en una temática más específica como es la medicina y la diferencia entre cada de estas estiquetas es la especialidad médica de las que trata.

4.2. Selección de parámetros de simulación

A partir de la descripción de los algoritmos de *clustering* espectral a evaluar se lleva a cabo su programación en Matlab. Cada uno de estos algoritmos ha sido diseñado y programado en función de unos parámetros de entrada, que van a ser descritos a continuación particularizando para cada algoritmo, y fijando como resultado o salida de estos algoritmos el *clustering* calculado y la medida de bondad de dicho agrupamiento basada en la información mutua, como se explicó en el capítulo 3.

4.2.1. Clustering en un espacio de características

Este algoritmo tiene como parámetros de entrada la matriz formada con las *bag of words* de los documentos de la base de datos que se esté estudiando

y el número de *clusters* en los que se desea agrupar o, en este caso, autoorganizar los documentos. Como resultado obtenemos el *clustering* calculado y el valor de información mutua que se obtiene al compararlo con el *clustering* real.

Como a priori no conocemos el número de *clusters* que genera el *clustering* con mejor información mutua, se realizan simulaciones variando el número de clusters de 2 a 50. Hemos fijado un valor máximo de *clusters* a 50 porque en aplicaciones reales no vamos a tener que organizar textos en un número de categorías muy elevadas.

Tomando como referencia la elección de parámetros y criterios de simulación, a continuación mostramos los resultados de la simulación del algoritmo de *clustering* espectral en un espacio de características para las tres bases de datos en estudio.

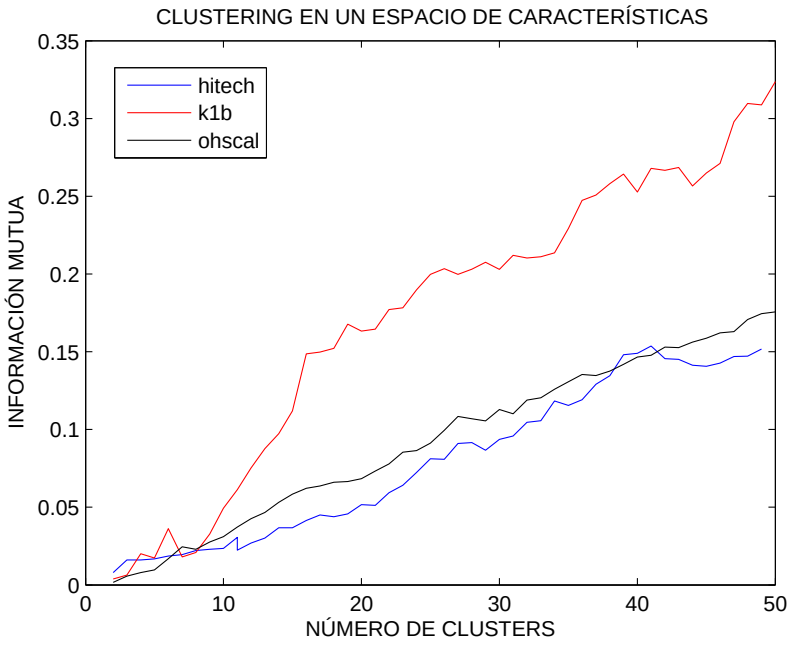

Figura 4.4: Resultados de información mutua de algoritmo de *clustering* espectral en un espacio de caractísticas para las tres bases de datos

Como podemos observar en la figura, la tendencia en los valores de información mutua en relación al número de *clusters* es creciente. Este comportamiento es el esperado, puesto que cuantos más *clusters* tenga el agrupamiento, cada uno de ellos concentrará menos documentos y éstos serán más parecidos, aumentando el valor de información mutua.

4.2.2. Cortes normalizados

El algoritmo de cortes normalizados además de recibir como parámetro de entrada el número de *clusters* y la matriz de *bag of words* de los documentos necesita saber qué tipo de *splitting point* tiene que calcular en el proceso de partición del grafo que representa a los documentos.

Como vimos en el capítulo 2, exiten cuatro formas de selección del *splitting point*:

- Asignarle al *splitting point* el valor 0.
- Calcular la media de los valores de las componentes del autovector y asignarle ese valor.
- Calcular la mediana de los valores de las componentes del autovector y asignarle ese valor.
- Seleccionar el *splitting point* que mejor corte normalizado genere.

Cada uno de estos procedimientos de elección del parámetro se representa con un código numérico que se introduce por parámetro al programa.

Al igual que el algoritmo anterior, se pretende estudiar los valores de información mutua variando el número de *clusters*, por consiguiente se realizan simulaciones variando este parámetro de 2 a 50 *clusters*.

A continuación podemos observar el resultado de dichas simulaciones estudiando para cada base de datos qué método de elección de *splitting point* genera mejores resultados.

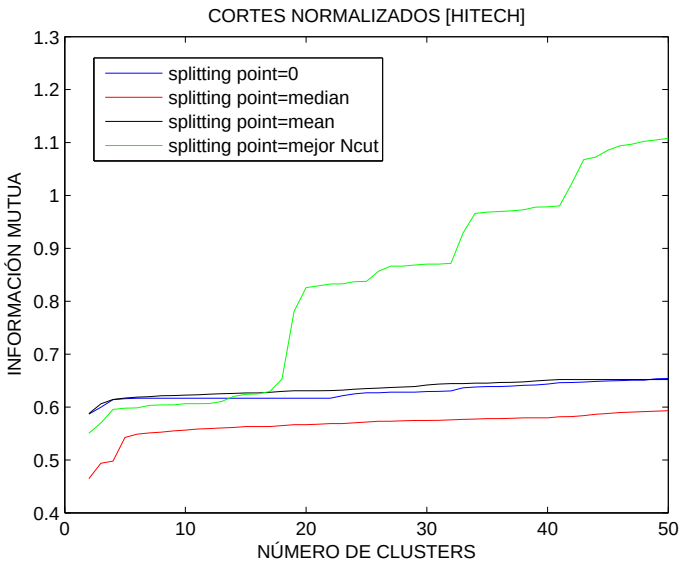

Figura 4.5: Resultados de información de algoritmo de Corte Normalizado para *hitech*

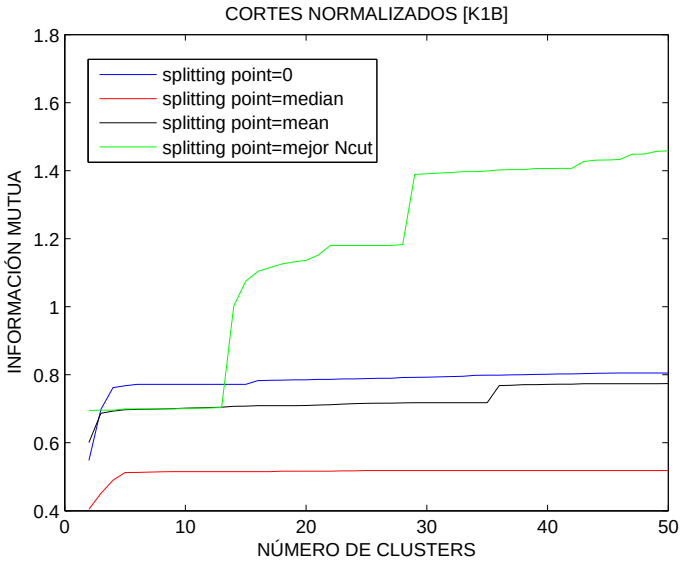

Figura 4.6: Resultados de información mutua de algoritmo de Corte Normalizado para *k1b*

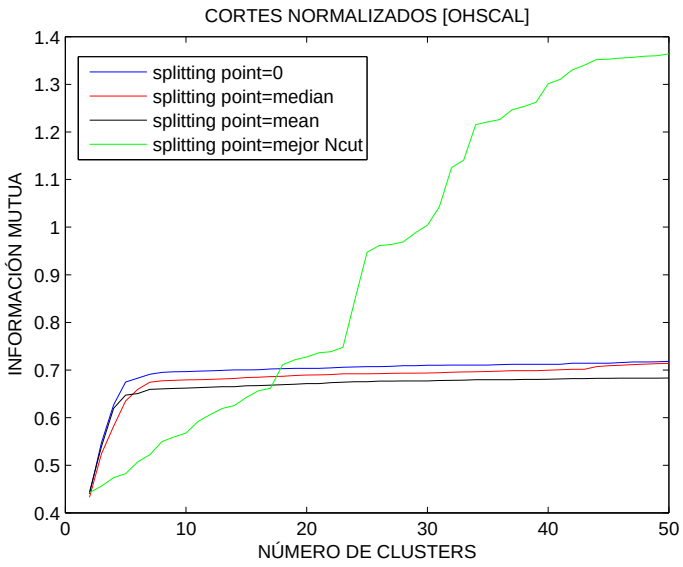

Figura 4.7: Resultados de información de algoritmo de Corte Normalizado para *ohscal*

Para cada base de datos seleccionamos el mejor método de elección de *splitting point*, es decir, el que genera mejores resultados de información mutua. Para las tres bases de datos, se consiguen mejores resultados de información mutua seleccionando el *splitting point* con el cuarto método, es decir seleccionando aquel *splitting point* que mejor corte normalizado genere. A continuación representamos esta selección realizada para cada base de datos.

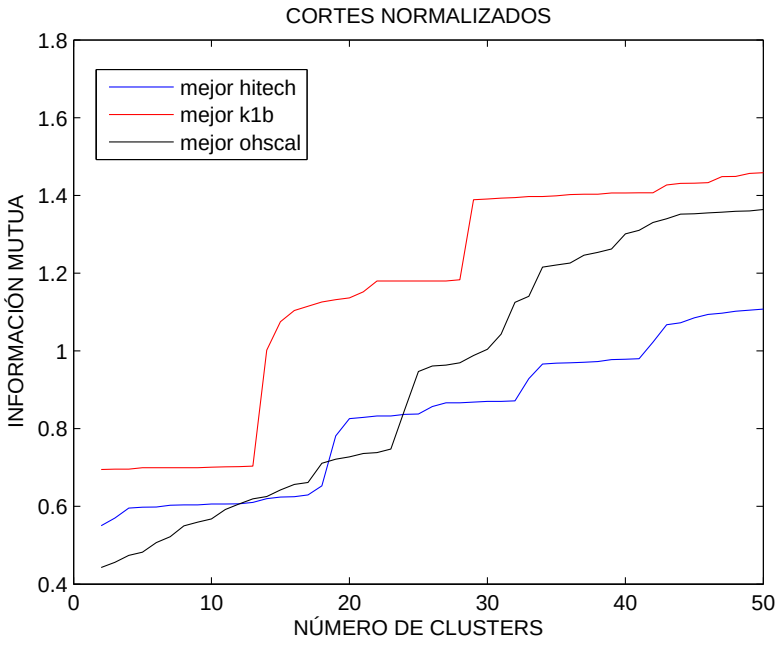

Figura 4.8: Resultados de información mutua de algoritmo de Corte Normalizado para las tres bases de datos

4.2.3. K-medias ponderado

Como hemos visto en el capítulo 2, el algoritmo de K-medias ponderado dispone de dos variantes en la función de coste asociada a la realización del clustering. Por tanto, como parámetros de entrada al algoritmo, además del número de *clusters* y la matriz de *bag of words* tenemos que indicar cuál de las funciones de coste queremos utilizar.

Debido a la inicialización aleatoria de los centroides en el K-medias, se realizan 10 iteraciones dentro de cada simulación, para obtener un resultado estadísticamente estable, calculando la media de los valores de información mutua conseguidos en esas iteraciones.

A continuación se muestran los resultados de información mutua para cada base de datos, realizando simulaciones para dos funciones de coste, variando el número de clusters de 2 a 50.

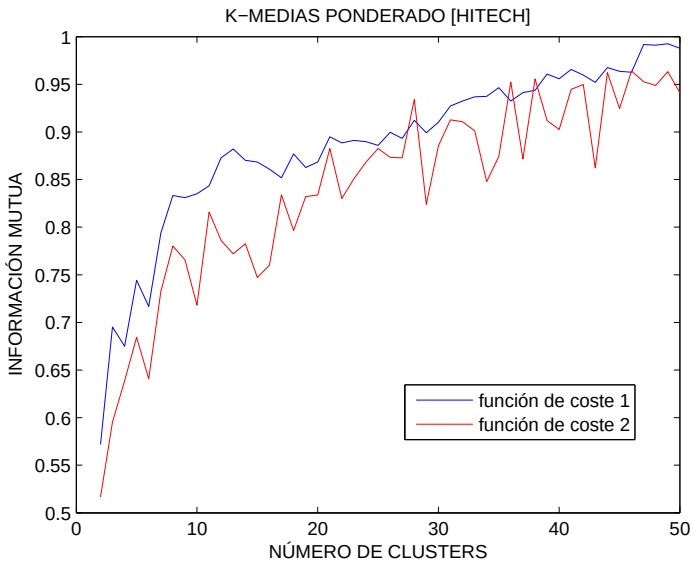

Figura 4.9: Resultados de información mutua de algoritmo de K-medias ponderado para *hitech*

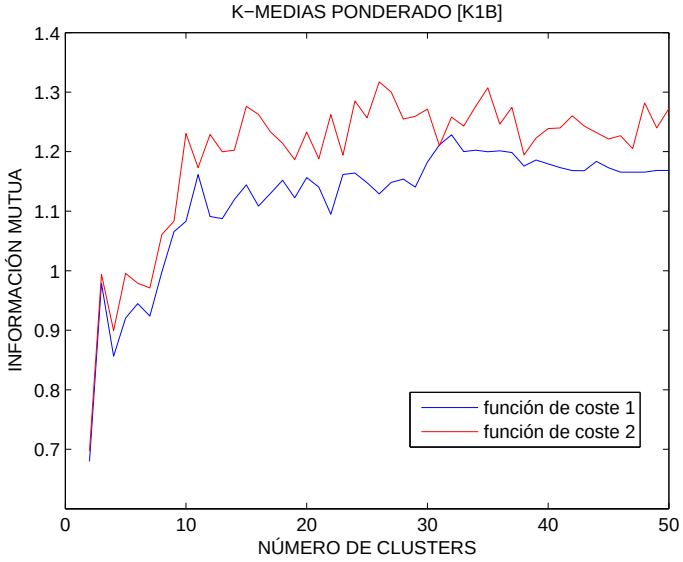

Figura 4.10: Resultados de información mutua de algoritmo de K-medias ponderado para *k1b*

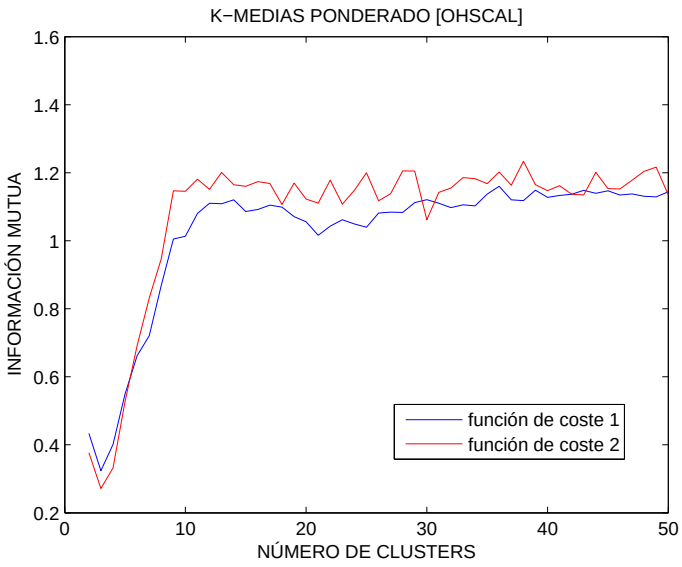

Figura 4.11: Resultados de información mutua de algoritmo de K-medias ponderado para *ohscal*

Como hemos visto en las figuras anteriores, para la base de datos *hitech* se obtienen mejores resultados usando la función de coste 1, mientras que para *k1b* y *ohscal* la función de coste 2 consigue mejores valores de información mutua.

Si nos fijamos en la forma de las figuras, vemos que oscila bastante, más que para el resto de algoritmos. Se trata de un algoritmo que posee más aleatoriedad que el resto, debido a la inicialización aleatoria de los centroides. Aún así, nos interesa la tendencia de estas figuras y al igual que en el resto de algoritmos, los valores de información mutua aumentan con el número de clusters.

A continuación se muestran los mejores resultados para cada base de datos.

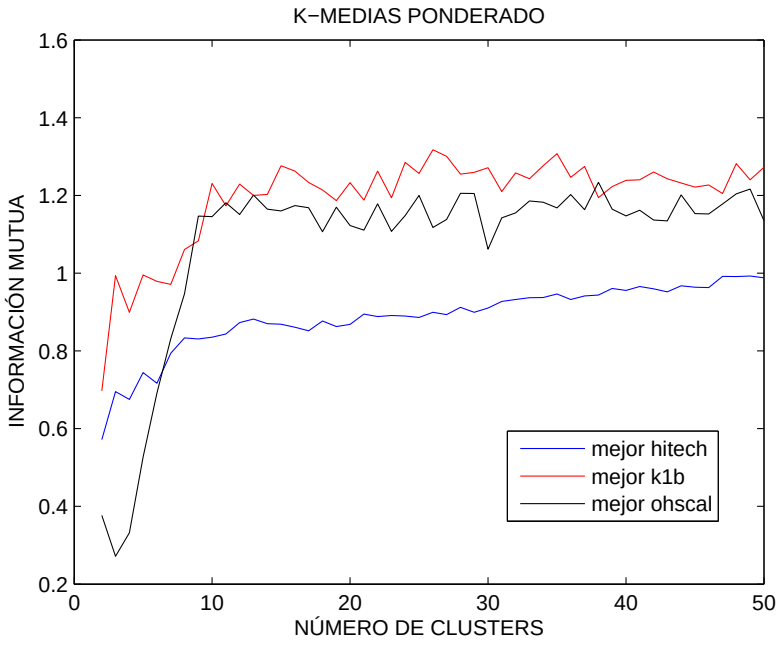

Figura 4.12: Resultados de información mutua de algoritmo de K-medias ponderado para las tres bases de datos

4.2.4. Clustering espectral refinado

Este algoritmo solo recibe como parámetro de entrada el número de *clusters* y la matriz que contiene las *bag of words* de cada documento. Realizamos diversas simulaciones variando el número de *clusters* de 2 a 50 para estudiar como varía la información mutua de los *clusterings* con respecto al número de *clusters*. A continuación mostramos los resultados obtenidos para las tres bases de datos.

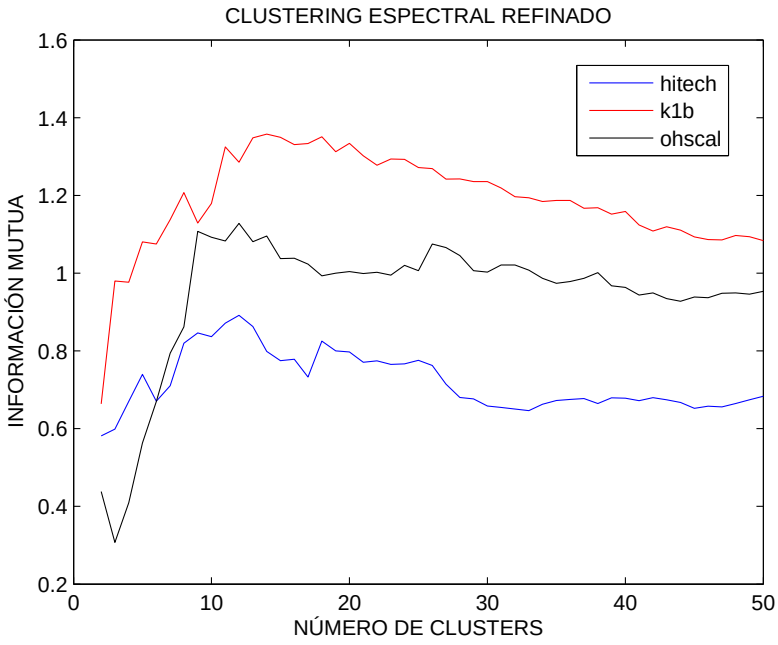

Figura 4.13: Resultados de información mutua de algoritmo de Clustering Espectral Refinado para las tres bases de datos

4.3. Comparación entre algoritmos

Una vez tenemos los resultados de las simulaciones de nuestros algoritmos de *clustering* a continuación vamos a mostrar una comparativa entre ellos para cada base de datos y seleccionando cuál de estos algoritmos nos resulta el más válido en la autoorganización de documentos.

Vamos a hacer esta comparación en función de la información mutua de los *clusterings* calculados para cada base de datos y también vamos a evaluar los resultados realizando un ejercicio de clasificación o etiquetado de los documentos a partir del *clustering* calculado y evaluar que tasa de acierto podemos conseguir con cada algoritmo.

4.3.1. Comparación basada en información mutua

A continuación mostramos los mejores resultados de cada algoritmo para cada base de datos y determinaremos cuál de ellos autoorganiza mejor los documentos de cada base de datos.

Comparación algoritmos para *hitech*

Si representamos los resultados de las simulaciones de los cuatro algoritmos sobre la base de datos *hitech* obtenemos la siguiente figura, donde además representamos el valor máximo de información mutua que puede alcanzarse a partir del *clustering* real. Como se ha explicado en el capítulo 3, el valor máximo de información mutua coincide con el valor de entropía de dicho agrupamiento y para este caso su valor es 2.4152.

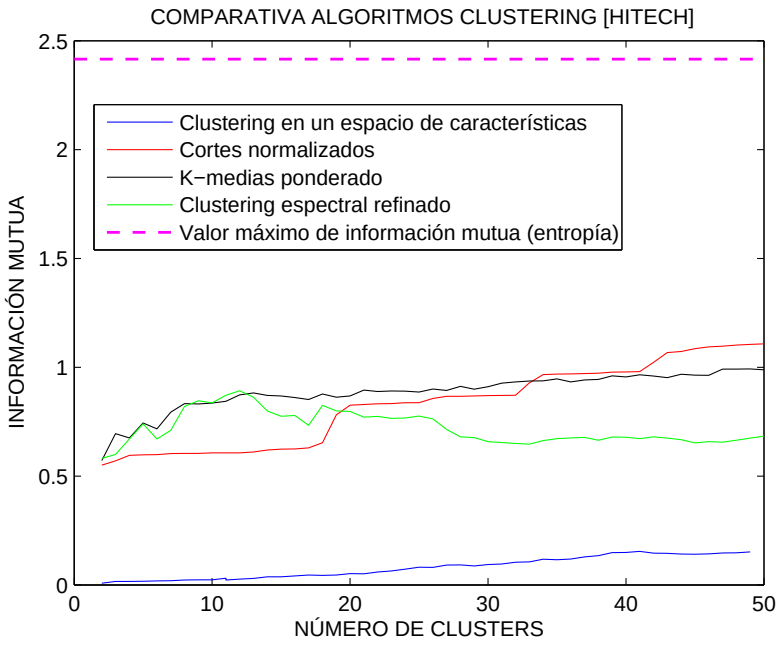

Figura 4.14: Resultados de información mutua para *hitech*

Calculando el valor medio de los resultados de información mutua para 2 hasta 50 clusters para cada uno de los algoritmos, obtenemos los siguientes resultados, donde podemos corroborar que el algoritmo de K-medias ponderado es el que mejores resultados obtiene para esta base de datos en la tarea de autoorganización de documentos.

Algoritmo	Valor medio
Clustering espacio características	0.0792
Cortes normalizados	0.8288
K-medias ponderado	**0.8871**
Clustering espectral refinado	0.7103

Cuadro 4.2: Valores medios de información mutua de cada algoritmo para base de datos *hitech*

Si comparamos el valor medio del algoritmo de K-medias ponderado con el valor máximo de información mutua, observamos que es un 36.73 % de dicho máximo, por tanto aun está lejos de conseguir llegar a un resultado óptimo, pero está claro que este algoritmo es el que más cerca se encuentra de este mejor valor de información mutua asociado a la entropía.

Comparación algoritmos para *k1b*

Al igual que para la base de datos anterior, representamos los resultados de los algoritmos calculados sobre la base de datos *k1b*, representando también su valor máximo de información mutua, que para este caso es de 1.1196.

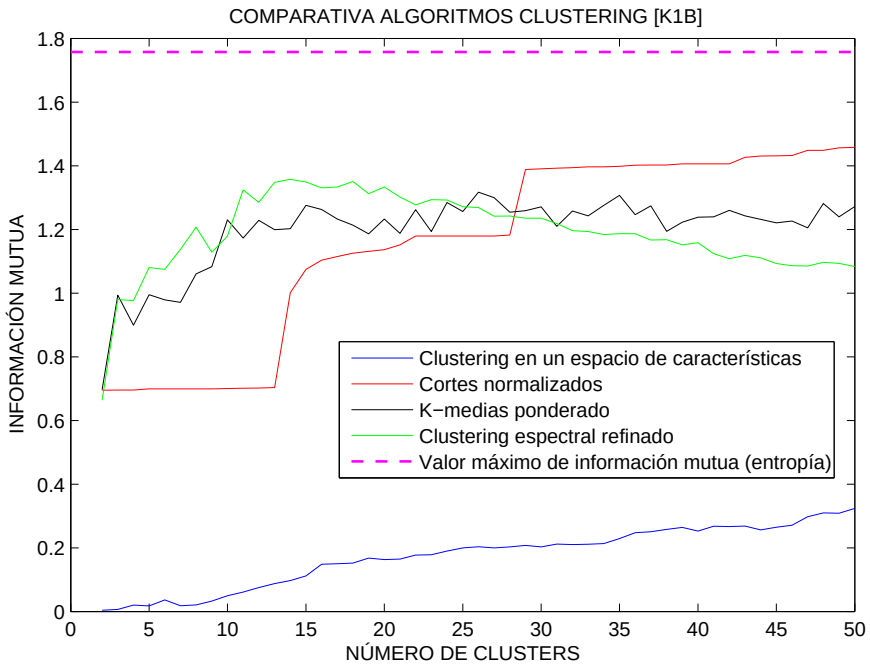

Figura 4.15: Resultados de información mutua para *k1b*

Calculamos el valor medio de los valores de información mutua para un número de clusters de 2 hasta 50, consiguiendo los siguientes resultados:

Algoritmo	**Valor medio**
Clustering espacio características	0.1741
Cortes normalizados	1.1556
K-medias ponderado	**1.196**
Clustering espectral refinado	1.1885

Cuadro 4.3: Valores medios de información mutua de cada algoritmo para base de datos *k1b*

Al igual que la base de datos anterior, el algoritmo de K-medias ponderado consigue los mejores resultados en la tarea de autoorganización de documentos. Si comparamos el valor medio que obtenemos con este algoritmo con el valor máximo de información mutua dado por la entropía del agrupamiento real, vemos que es un 63.69 % del valor óptimo. Proporcionalmente, el mismo

algoritmo para esta base de datos alcanza mejores resultados que en la base de datos anterior.

Comparación algoritmos para *ohscal*

Representamos los resultados de las simulaciones de los algoritmos para esta base de datos, representando también el valor máximo de información mutua (3.2705) para poder valorar los resultados.

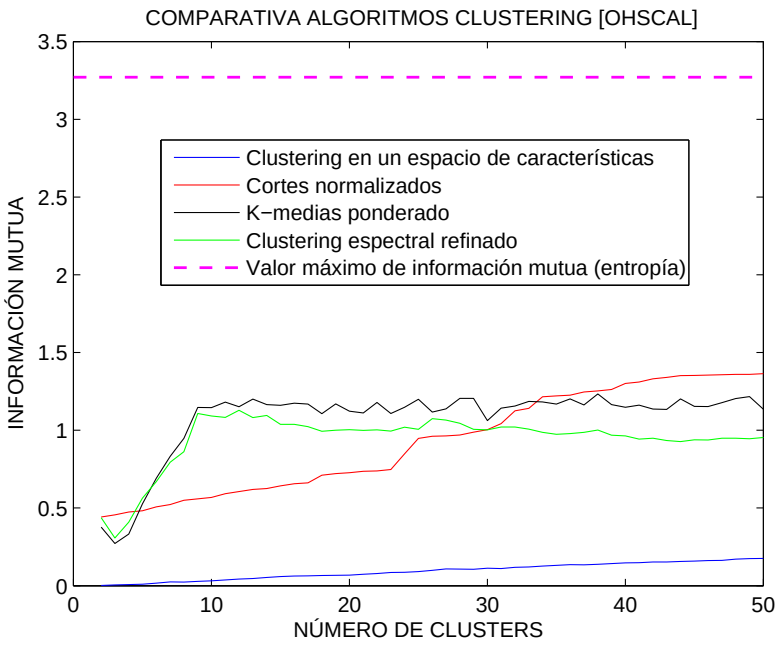

Figura 4.16: Resultados de información mutua para *ohscal*

Al igual que los casos anteriores, calculamos el valor medio de la información mutua calculada para un número de clusters desde 2 a 50.

Algoritmo	Valor medio
Clustering espacio características	0.0934
Cortes normalizados	0.9294
K-medias ponderado	**1.0767**
Clustering espectral refinado	0.9444

Cuadro 4.4: Valores medios de información mutua de cada algoritmo para base de datos *ohscal*

Una vez más el algoritmo de K-medias ponderado consigue los mejores resultados. Comparando de nuevo con el valor máximo de información mutua dado por la entropía, el valor medio es un 32.92 % de dicho máximo, con lo que proporcionalmente vemos que es un resultado bastante lejano del máximo y se aleja del calculado para la base de datos anterior, pero aun así es el mejor resultado posible entre todos los algoritmos.

4.3.2. Comparación basada en clasificación de documentos

Como hemos comentado en la descripción de las bases de datos, el archivo *.clabel contiene las etiquetas reales de los documentos. Nos hemos planteado desarrollar otro procedimiento para validar si nuestros algoritmos son capaces de organizar de manera correcta los documentos. A partir del *clusterings* calculado por nuestros algoritmos vamos a etiquetar a cada uno de los grupos con la etiqueta real mayoritaria de los documentos que pertenecen a dicho grupo y con esta etiqueta de grupo re-etiquetaremos a los documentos contenidos en él. De esta manera compararemos, para todos los documentos, si las etiquetas reales y las calculadas de esta forma coinciden, obteniendo así indicador que hemos denominado tasa de acierto.

A continuación mostramos los resultados obetenidos para cada base de datos, indicando que algoritmo consigue los mejores resultados.

Resultados del etiquetado para *hitech*

Los documentos de esta base de datos están etiquetados en base a 6 temáticas diferentes como se ha detallado en la descripción de la base de datos. Por tanto se calculan los *clusterings* de los documentos de esta base de

datos con los cuatro algoritmos para 6 clusters. Además se realizan *clusterings* con 9 y 12 grupos, puesto que en tareas de clasificación se suele trabajar con más grupos de los reales. Puesto que el número real de grupos es 6, se ha probado con un número de grupos igual a multiplicar por 1.5 este número inicial y por el doble.

A continuación mostramos los resultados en la tasa de acierto calculada de la manera que hemos explicado anteriormente. Se ha calculado esta tasa para los *clusterings* calculados con cada uno de nuestros cuatro algoritmos tomando 6, 9 y 12 *clusters*.

Algoritmo	**6 grupos**	**9 grupos**	**12 grupos**
Clustering espacio características	26.99 %	26.77 %	26.81 %
Cortes normalizados	46.15 %	46.15 %	46.15 %
K-medias ponderado	52.85 %	60.41 %	59.93 %
Clustering espectral refinado	52.02 %	**61.58 %**	60.45 %

Cuadro 4.5: Tasa de acierto para *hitech*

El mejor resultado de etiquetado en esta base de datos se consigue con el algoritmo de *clustering* espectral refinado para 9 *clusters*, obteniendo una tasa de acierto del 61.58 %. Hay que destacar que con el algoritmo de K-medias ponderado obtenemos un resultado muy cercano a este.

Resultados del etiquetado para *k1b*

En esta base de datos los documentos son etiquetados según 6 temáticas, por tanto al igual que la base de datos anterior, se calculan los *clusterings* para 6, 9 y 12 grupos y posteriormente se etiqueta cada grupo con la etiqueta mayoritaria y se re-etiquetan los documentos. A partir de ahi se calcula la tasa de acierto computando que documentos tienencoincidencia entre la etiqueta real y la calculada. A continuación se muestran los resultados obtenidos:

Algoritmo	6 grupos	9 grupos	12 grupos
Clustering espacio características	59.36 %	60.13 %	60.13 %
Cortes normalizados	80.04 %	80.04 %	80.04 %
K-medias ponderado	82.05 %	88.76 %	**90.77 %**
Clustering espectral refinado	81.58 %	83.29 %	90 %

Cuadro 4.6: Tasa de acierto para *k1b*

El mejor resultado se obtiene con el algoritmo de K-medias para 12 *clusters*, con una tasa de acierto de 90.77 %, pero hay que destacar que se obtiene un resultado muy parecido con el algoritmo de *clustering* espectral refinado para 12 *clusters*.

Resultados del etiquetado para *ohscal*

Esta base de datos tiene 10 temáticas diferentes, por tanto se han calculado los *clusterings* para 10, 15 y 20 *clusters* utilizando los cuatro algoritmos que son objeto de estudio. Se han seleccionado 15 y 20 clusters por el mismo razonamiento utilizado anteriormente, porque se suelen realizar ejercicios de clasificación con más grupos que los reales y se ha tomado 1.5 por el número de grupos real y el doble de dicho número de *clusters*. A continuación se muestran los resultados obtenidos en la tasa de acierto en la tarea de etiquetado:

Algoritmo	10 grupos	15 grupos	20 grupos
Clustering espacio características	15.16 %	16.40 %	17.16 %
Cortes normalizados	25.48 %	27.04 %	30.24 %
K-medias ponderado	**49 %**	47.68 %	44.36 %
Clustering espectral refinado	46.56 %	44.12 %	41.24 %

Cuadro 4.7: Tasa de acierto para *ohscal*

El mejor resultado de etiquetado lo conseguimos con el algoritmo K-medias ponderado para 10 clusters. El valor de esta tasa es más bajo que los conseguidos para las otras dos bases de datos.

4.4. Análisis semántico

Como vimos en la descripción de las bases de datos, cada una de ella tiene propiedades que repercuten de una u otra manera en los resultados proporcionados por los algoritmos. Una de esas propiedades es el contexto semántico de cada una de las bases de datos y la repercusión que las temáticas y por tanto las palabras que forman cada documento tiene sobre nuestros resultados. Este hecho hace resaltar para qué tipo de bases de datos nuestros algoritmos resultan más eficientes y para qué otras no lo son.

A continuación mostramos qué consecuencias ha tenido este aspecto en los resultados obtenidos para cada base de datos.

4.4.1. *Hitech*

Los documentos de esta base de datos poseen temáticas que contienen palabras comunes a unas y a otras, lo que genera que al comparar documentos y agruparlos mediante nuestros algoritmos, algunos documentos caigan en grupos en los que su temática difiere de la original. Esto queda de manifiesto cuando analizamos las palabras más usadas en los documentos pertenecientes a cada grupo, observando la coincidencia de muchas palabras comunes entre ellos. Si observamos las palabras más utilizadas en los documentos pertenecientes a las etiquetas *computers*, *electronics* y *technology*, corroboramos este hecho, puesto que palabras como *comput*, *compani*, *market*, *million*, *percent*, *price*, *product*, *program*, *sale*, *system* o *time*, tienen gran repetitividad en esos tres grupos. También existe mucha relación entre los documentos con etiqueta *health* y *medical*, donde palabras como *care*, *center*, *doctor*, *health*, *hospit*, *medic*, *patient* y *people* son comunes a ambos grupos.

4.4.2. *K1b*

Los resultados en esta base de datos son mejores que en las otras debido a dos factores: el primero, como vimos en la descripción de la base de datos, la gran mayoría de los documentos pertenecen a la temática *entertainment* con lo que en el ejercicio de etiquetado es menos probable equivocarse en la re-etiquetación de los documentos puesto que a priori es más probable que los textos tengan etiqueta *entertainment* que cualquier otra. El segundo factor es debido a que las temáticas de los documentos de esta base de datos son más diferentes entre sí o al menos no tan relacionados como en las otras bases de

datos, lo que lleva a que cuando realizamos la comparación entre documentos, éstos sean más diferentes entre sí y la clasificación sea más eficiente.

4.4.3. *Ohscal*

Los resultados que hemos obtenido para esta base de datos han sido peores que para las otras dos y el motivo puede deberse precisamente a la relación semántica entre los documentos de los grupos. Todos las temáticas de esta base de datos están relacionadas con la medicina y cada una de ellas está relacionada con un campo o un aspecto específico de la medicina, pero las palabras que conforman los documentos son muchas veces comunes entre sí, lo que conlleva a que la comparación entre documentos no sea tan diferenciada cuando se trata de temáticas diferentes. Observando algunas de las palabras más utilizadas en los documentos pertenecientes a las diferentes etiquetas podemos comprobar este hecho. Todos los grupos utilizan palabras comunes ligadas a la medicina, como *cell*, *human*, *infect*, *patient*, *protein*, *tumor*, *cancer*, *result*, *treatment*, *clinic* o *cancer*. Por tanto, esta base de datos pone de manifiesto que nuestros algoritmos son menos eficientes cuando tratamos con temáticas poco diferenciadas entre sí, puesto que la compartición de palabras comunes provoca fallos en el *clustering*.

Capítulo 5

Conclusiones

El proyecto se planteaba en el Capítulo 1 unos objetivos basados en el análisis de las prestaciones de los algoritmos de *clustering* espectral para el problema de autoorganización de documentos. Para ello contextualizamos nuestra tarea dentro de un problema de minería de datos y una vez descritas cada una de las técnicas de *clustering* espectral a implementar y la parametrización de los documentos a autoorganizar, se realizaron los experimentos necesarios para poder llegar al objetivo marcado de determinar cuál de dichos algoritmos resulta ser el más óptimo en nuestra tarea.

A partir de los experimentos realizados sobre las tres bases de datos de documentos, *hitech*, *k1b* y *ohscal*, concluimos que de los cuatro algoritmos de *clustering* espectral, el método de K-medias ponderado es el que mejor resultados obtiene para las tres bases de datos en base a valores de información mutua y tasa de acierto conseguidas en el re-etiquetado de documentos basado en *clustering*. Por tanto, puesto que las tres bases de datos poseen características internas diferentes podemos decir que dicho método es un método idóneo para la autoorganización de documentos, siendo las bases de datos de diferentes perfiles, temáticas o estructuras. Este algoritmo ha resultado ser el más robusto en el cómputo global de los experimentos realizados.

Un asunto importante a tener en cuenta es la sensibilidad de estos algoritmos de *clustering* espectral con la estructura interna de las bases de datos así como las características de los documentos. Como hemos visto, en el caso concreto de la base de datos *k1b*, cuando los documentos tienen temáticas claramente diferenciadas entre sí, los algoritmos estudiados obtienen resultados mucho mejores que en el caso de bases de datos en los que las temáticas son más cercanas unas de otras y por tanto, los documentos pueden contener palabras comunes a varias temáticas. Este hecho hace que en la compara-

ción de documentos, la medida de similitud sea mayor de lo que debería a la vista de las diferentes temáticas a las que pertenecen. Como hemos visto en la base de datos *k1b*, documentos pertenecientes a temáticas diferenciadas hacen que los resultados estén más cerca del valor máximo alcanzable, que para el caso de bases de datos con temáticas más cercanas. Cuando hemos evaluado los *clusterings* con la información mutua, hemos visto que las prestaciones para *k1b* están muy cercanas al valor máximo de entropía que puede conseguir. Hemos podido contrastar este asunto cuando hemos realizado el ejercicio de etiquetado, calculando las tasas de acierto, puesto que hemos alcanzado valores de esta medida muy altos y por tanto muy satisfactorios para la resolución de nuestra tarea. El problema es que en ejercicios reales de autoorganización de documentos, las temáticas bien diferenciadas son muy complicadas de conseguir.

Por ello una de las posibles líneas futuras a desarrollar como mejora del proyecto realizado sería encontrar un criterio de comparación entre documentos mucho más sensible a este tema. Está claro que este asunto es clave para la obtención de mejores resultados en la autoorganización de documentos, por tanto desarrollando una medida de similitud más compleja, que diferencie mucho más los documentos entre sí cuando ocurra este problema. Avanzando un poco más en esta línea, sería muy interesante incorporar un análisis lingüistico más profundo en la comparación de documentos. Por ejemplo, a la hora de comparar dos documentos, es posible que siendo de temáticas parecidas, por el hecho de utilizar sinónimos, metáforas o estructuras lingüisticas más complejas, en una comparación básica puede que resulten ser documentos diferentes. Por tanto, es interesante desarrollar sistemas de análisis de documentos y comparación entre ellos basándonos en técnicas lingüísticas para abordar este problema de una manera más completa y eficiente.

Por otro lado, sería interesante desarrollar un método de evaluación de la autoorganización de documentos más específico para este problema. En este proyecto nos hemos basado en la información mutua aportada por el *clustering* calculado comparándolo con el *clustering* real. Este método es válido para cualquier tipo de dato de entrada, puesto que no requiere saber la naturaleza de los datos, simplemente compara agrupaciones entre sí. Realizando el ejercicio de etiquetado, nos hemos acercado un poco más al contexto particular del problema, puesto que sí que tenemos en cuenta que los datos agrupados son documentos y que éstos tienen una etiqueta asociada. Sería muy útil por tanto dar un paso más e intentar realizar un análisis más profundo a la estructura semántica de los datos, evaluando con mucho más detalle las palabras que componen los documentos y comprobando que efectivamente los documentos de un mismo grupo poseen semánticas comunes. En este pro-

yecto se ha intentado desarrollar alguna comprobación de este tipo, aunque a menor escala, pero resultaba inviable debido al tamaño de los diccionarios y al tamaño de los grupos para los que se han calculado los agrupamientos, por tanto quedaba fuera de lo objetivos del proyecto.

Queda de manifiesto la complejidad en la resolución de este problema, fundamentalmente por lo complejo que resulta la propia estrutura del contexto en estudio. Cuando trabajamos con documentos escritos, resulta complicado determinar de manera automática la semántica de éstos, puesto que los seres humanos nos comunicamos de diferentes maneras, dobles sentidos, metáforas, etc. Por tanto, es un reto futuro, no sólo para este problema sino para muchos otros, conseguir mejorar el proceso de automatización de temas relacionados con la comunicación entre humanos.

A pesar de que la autoorganización de documentos es una tarea compleja, mediante la realización de este proyecto hemos mostrado soluciones para resolverlo basadas en algoritmos de *clustering* espectral. A partir del estudio realizado, concluimos que el algoritmo de K-medias ponderado es el que mejores resultados consigue. En la resolución del problema ha quedado de manifiesto la importancia de la organización de los documentos de las bases de datos en cuanto a etiquetado se refiere. De ahí que se hayan propuesto una serie de líneas futuras a desarrollar para así conseguir una resolución más completa y robusta al problema de autoorganización de documentos mediante algortimos de *clustering* espectral.

Apéndice A

PRESUPUESTO DEL PROYECTO

En este apéndice se presentan los costes de la realización de este Proyecto Fin de Carrera. En la siguiente tabla se muestran las fases del proyecto y la duración aproximada de cada una de esas fases.

Fase del proyecto	**Nº de horas**
Documentación	375h
Desarrollo software	140h
Simulaciones	250h
Redacción de la memoria	260h

Cuadro A.1: Fases de desarrollo del proyecto fin de carrera

Por tanto, el tiempo aproximado de dedicación al desarrollo de este proyecto es de 1025h. Teniendo en cuenta que los honorarios que percibe un Ingeniero de Telecomunicación son de 60 euros la hora, el coste personal del proyecto es de 61500 euros.

Para poder llevar a cabo el trabajo experimental, fue necesaria la adquisición de un ordenador de gran capacidad puesto que con bases de datos tan grandes se requería un ordenador con bastante memoria RAM y con mucha capacidad de procesador. El coste total de este ordenador fue de unos 1000 euros.

Como coste material también hay que añadir el coste del programa Matlab con el que se programaron los algoritmos y se realizaron las simulaciones. El coste de este programa es de unos 100 euros.

Por tanto, el coste total del proyecto fin de carrera es de 62600 euros.

Bibliografía

[1] Data clustering software. karypis lab. cluto. software for clustering high-dimensional datasets. http://glaros.dtc.umn.edu/gkhome/views/cluto/.

[2] F. Bach and M. Jordan. *Learning Spectral Clustering, With Application to Speech Separation. Journal of Machine Learning Research*, 7:1963–2001, 2006.

[3] R. Duda and P. Hart. *Pattern Classification.* Wiley-Interscience Publication, 2000.

[4] T. Joachims. *Learning to Classify Text using Support Vector Machines. Methods, Theory, and Algorithms.* Kluwer Academic Publishers, 2002.

[5] A. Ng, M. Jordan, and Y. Weiss. *On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems*, 2001.

[6] F. M. Reza. *An Introduction to Information Theory. Dover Publications*, 1994.

[7] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[8] J. Shi and J. Malik. *Normalized Cuts and Image Segmentation. Transations on Pattern Analysis and Machine Intelligence, IEEE*, 22(8), April 2000.

[9] L. Zelnik-Manor and P. Perona. *Self-Tuning Spectral Clustering.* http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html.

[10] X. Zhu and I. Davidson. *Knowledge Discovery and Data Mining: Challenges and Realities.* Information Science Reference, 2007.